VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky

DIPLOMOVÁ PRÁCE

2015 Bc. Tomáš Urbanczyk

VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra kybernetiky a biomedicínského inženýrství

Návrh a dílčí implementace elektronické verze Záznamu Intenzivní Péče

Design and Partial Implemetation of e-version of Intesive Care Record

2015 Bc. Tomáš Urbanczyk

Zadání

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra kybernetiky a biomedicínského inženýrství

Zadání diplomové práce

Student: **Bc. Tomáš Urbanczyk**

Studijní program: N2649 Elektrotechnika

Studijní obor: 3901T009 Biomedicínské inženýrství

Téma: Návrh a dílčí implementace elektronické verze Záznamu Intenzivní Péče
Design and Partial Implemetation of e-version of Intesive Care Record

Zásady pro vypracování:

Tématem diplomové práce je analýza a návrh možností IT podpory pro pořizování Záznamu Intenzivní Péče (dále ZIP) na jednotce urgentního příjmu FN Ostrava s cílem optimalizovat tento proces z hlediska jeho úplnosti, spolehlivosti, bezpečnosti i rychlosti. Dalším cílem práce je funkce systému identifikované v procesu analýzy implementovat a jejich implementaci pilotně ověřit v provozu urgentního příjmu. Tento krok předpokládá udělení souhlasu vedení pracoviště urgentního příjmu. Technické řešení tématu bude provedeno na mobilní platformě s OS Android a Windows platformě pomocí technologie .NET.

Postup řešení:
1. Analýza provozu a procesního postupu na urgentním příjmu a Záznamu Intenzivní Péče.
2. Návrh funkcí a způsobu jejich implementace a rozdělení na předpokládané platformy.
3. Implementace vybraných funkcí.
4. Testování v provozu urgentního příjmu.
5. Zhodnocení procesu testování a zhodnocení možnosti komplexního řešení systému pro podporu pořizování Záznamu Intenzivní Péče.

Seznam doporučené odborné literatury:

[1] MURPHY, Mark L. *Android 2 Průvodce programováním mobilních aplikací.* překlad Jakub MUŽÍK. 1.vyd. Brno:COMPUTER PRESS, 2011. ISBN 978-80-251-3194-7.
[2] MEIER, Reto. *Proffesional Android Application Development.* Indianopolis(USA): Wiley Publishing Inc., 2009. ISBN 978-0-470-34471-2.
[3] ECKEL, Bruce. *Myslíme v jazyku Java: knihovna programátora.* Praha:Grada Publishing, 2001. ISBN 80-247-9010-6.
[4] ANDROID DEVELOPERS. The Developer's Guide [online][cit. 2012-10-12]. Dostupné také z: http://developer.android.com/guide/index.html.
[5] URBANCZYK, Tomáš. *Databáze pro evidenci označených entit sledující koncept Internet of Things.* Ostrava, 2013. Bakalářská práce. VŠB - TU Ostrava, Fakulta elektrotechnika a informatiky. Vedoucí práce Ing. Jakub Jirka.

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **doc. RNDr. Jindřich Černohorský, CSc.**

Datum zadání: 01.09.2014

Datum odevzdání: 07.05.2015

doc. Ing. Jiří Koziorek, Ph.D.
vedoucí katedry

prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

Datum odevzdání: 7. 5. 2015

Bc. Tomáš Urbanczyk

Poděkování

Děkuji panu doc. RNDr. Jindřichu Černohorskému, CSc. vedoucímu mé diplomové práce za cenné rady, čas a připomínky, které mi v průběhu psaní poskytnul.

Dále děkuji Mgr. Jindřišce Kosinové pracovnici urgentního příjmu za výbornou komunikaci a pomoc, které se mi dostalo.

Děkuji Mgr. Pavlíně Štěpánové za výbornou spolupráci a odborné rady.

Abstrakt

Cílem této diplomové práce je navrhnout informační systém, který bude schopný evidovat záznamy intenzivní péče na oddělení urgentního příjmu fakultní nemocnice Ostrava. Realizován bude na mobilním zařízení s platformou Android a technologii NFC vycházející z konceptu internet of Things. Obsahem práce je analýza prostředí s popisem stávajícího řešení. Dále návrh, implementace a testování informačního systému.

Diplomová práce je tvořena v Android Studiu verze 1.1.0. Testovací mobilní zařízení značky Samsung Galaxy S4 (GT-I9505) s verzí Android 4.4.2. Databáze je spravována pomocí MySQL Workbench 6.3 CE

Klíčová slova

Android, Android Studio, Databáze, Identifikační technologie, urgentní příjem FNO, Informační systém

Annotation

The aim of this master thesis is to design an information system, which will be able to register the records of an intensive care unit of emergency department of the faculty hospital of Ostrava. It will be implemented on a mobile device with an Android platform with a NFC technology based on the Internet of Things concept. The content of the project is an analysis of environment with a description of current solution, as well as design, implementation and testing of the information system.

The master thesis is created in Android Studio, version 1.1.0. Testing mobile device, type Samsung Galaxy S4 (GT-I9505) with 4.4.2 Android version. The database is administered with MySQL Workbench 6.3 CE.

Key words

Android, Android Studio, Database, Identification technology, emergency department of the faculty hospital of Ostrava, Information system

Seznam použitých zkratek

FNO	Fakultní Nemocnice Ostrava
EAN	European Article Number, Evropský číselný kód
QR	Quick Response, Kód rychlé reakce
RFID	Radio Frequency Identification, Rádio frekvenční komunikace
NFC	Near Field Communication, pole blízké komunikace
SQL	Structured Query Language, Strukturalizovaný jazyk
API	Application Programming Interface
ERD	Entity-relationship model

Obsah

1 Úvod ... 1
1.1 Cíle diplomové práce ... 1
2 Internet věcí ... 2
2.1 Identifikační technologie ... 2
2.1.1 RFID ... 2
2.1.2 Více rozměrné tágy ... 5
2.1.3 QR kód ... 5
2.1.4 EAN ... 6
3 Android ... 7
3.1 Historie Android ... 7
3.2 Struktura androidu ... 7
3.3 Android Studio ... 8
3.3.1 Uživatelské rozhraní ... 8
3.3.2 Aplikace v Android Studiu ... 9
4 Uložiště ... 10
4.1 Databáze MySQL ... 10
4.1.1 Relační databáze ... 10
4.2 Interní databáze SQLite ... 12
4.2.1 Vytvoření a aktualizace s třídou SQLiteOpenHelper ... 12
5 Popis projektu ... 13
5.1 Stávající řešení ... 13
5.1.1 Příjem pacienta ... 13
5.1.2 Záznam intenzivní péče ... 15
5.1.3 Ukončení záznamu pacienta ... 16
5.2 Navrhované řešení ... 17
6 Analýza ... 18
6.1 Požadavky ... 18
6.1.1 Požadavky na techniku ... 18
6.2 Datová analýza ... 18
6.2.1 Specifikace dat ... 18
6.2.2 Lineární zápis dat ... 19
6.2.3 Datový slovník ... 20
6.2.4 ERD model dat ... 21
6.3 Funkční analýza ... 22

6.4 Návrh mobilní aplikace ... 23
6.4.1 Návrh uživatelského prostředí ... 23
6.4.2 Funkcionalita ... 25
6.5 Analýza identifikátoru ... 26
6.5.1 Typy identifikátoru ... 26
7 Technické řešení ... 27
7.1 Čtení identifikačních kódu ... 27
7.1.1 Zixing knihovna ... 27
7.1.2 AsyncTask ... 28
7.1.3 NFC ... 28
7.2 Ukládání dat ... 30
7.2.1 SQLite ... 30
7.2.2 Externí databáze ... 32
7.2.3 ListView ... 35
7.2.4 NavigationDrawer ... 36
8 Testování ... 37
8.1 Zhodnocení ... 38
8.1.1 Porovnání dokumentací ... 38
8.1.2 Chyby ... 38
9 Závěr ... 40

1 Úvod

Vývoj chytrých informačních nemocničních systémů v dnešní době nabírá nových rozměrů. S vývojem technologie rostou možnosti, kterými lze změnit dosavadní pohled na zdravotnickou dokumentaci. Z papírových záznamů se často přechází na elektronickou podobu. Tomu se tak děje z důvodu výhod, jenž elektronická dokumentace disponuje (např. kapacita úložného prostoru, rychlého vyhledávání, sdílení informací, přístup k informacím). Diplomová práce byla zadána na popud zvyšování kvality péče na urgentním příjmu fakultní nemocnice Ostrava (FNO), která se snaží pomocí nejnovějších trendů zefektivnit práci na oddělení a tím zvýšit bezpečnostní standart pro pacienta. Práce navazuje na bakalářskou práci s názvem „Databáze pro evidenci označených entit sledující koncept Internet of Things“. Tato práce vytvořila databázi schopnou evidovat data, která se shromažďují během pobytu pacienta na urgentním příjmu FNO. Další práce, která úzce souvisí s modernizací urgentního příjmu, je lokalizace pacienta v prostorách oddělení urgentního příjmu. Myšlenkou do budoucna je vytvořit informační systém, který by byl kompaktní, intuitivní a efektivní.

Zadáním diplomové práce je navrhnout a implementovat elektronickou verzi záznamu intenzivní péče (ZIP). To znamená nahradit dosavadní papírovou formu za elektronický systém založený na principech internet of things. Práce bude realizována na mobilním zařízení, které bude schopné načítat identifikační kódy a shromažďovat informace o průběhu léčby pacienta na urgentním příjmu. Výstupem tedy bude přehledný časový záznam intenzivní péče pacienta, který bude svým obsahem standardně vedený, jako dosud papírová forma, až na to, že data budou strukturalizována (schopna dalšího počítačového zpracovávání). Strukturalizace vede například, ke kontrole podávání léčiv, k logistice materiálu nebo ke kvalifikaci nestandartních postupů.

1.1 Cíle diplomové práce

- Analýza provozu a procesního postupu na urgentním příjmu a Záznamu Intenzivní Péče.

- Návrh funkcí a způsobu jejich implementace a rozdělení na předpokládané platformy.

- Implementace vybraných funkcí.

- Testování v provozu urgentního příjmu.

- Zhodnocení procesu testování a zhodnocení možnosti komplexního řešení systému pro podporu pořizování Záznamu Intenzivní Péče.

2 Internet věcí

Internet věcí, anglicky Internet of Things, je moderní technologie, která interaguje okolní prostředí s internetem. Jedná se o vnímání parametrů měřených snímači a následnou komunikací s řídící jednotkou. Řídící jednotka nebo-li centrála vyhodnocuje data a předkládá informace o okolním světě. V praxi to znamená použití identifikačních technologií na označení entity (zkoumaného objektu). Podle použité technologie můžeme informace o entitě sledovat, evidovat.

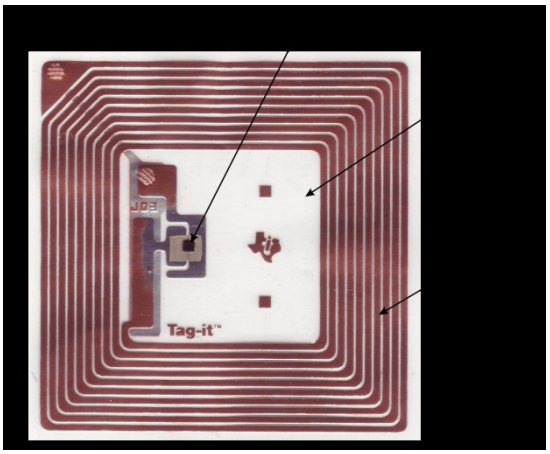
Internet věcí ve zdravotnickém informačním systému znamená evidování veškerých úkonů, které se vztahují k pacientovi. Pacient je, tedy zkoumaná entita (sledujeme např. fyziologické parametry, vyšetření, lékařské výkony, léky atd.). Již z výčtu parametru je jasné, že se pro evidování záznamu parametrů nebudou používat stejné identifikátory ale celá škála identifikátoru. To vede, k vysokým nárokům na čtecí zařízení.

2.1 Identifikační technologie

Identifikační technologie slouží k jednoznačnému určení entity a k výraznému zvýšení efektivní práce. Vývoj technologii značně ovlivňuje trendy tohoto odvětví. Dnes používáme identifikátory RFID vycházející z technologie NFC. Tyto identifikátory se dělí na aktivní a pasivní podle toho, jestli jsou napájeny z vlastního zdroje (aktivní), nebo takové které se nabijí přes anténu skrz rádiové vlnění (pasivní). Dále se vyskytují identifikátory charakterizované tvarem. To jsou například QR kódy nebo velmi známe čárové kódy (EAN).

2.1.1 RFID

RFID tag (obrázek č. 1) slouží podobně, jako čárové kódy k identifikaci, ale pracuje na principu radiofrekvenčního systému pomocí elektromagnetických vln. Skládá se z antény, čipu a propojení mezi nimi.[1] Jsou schopné zaznamenávat, uchovávat, pracovat v reálném čase.

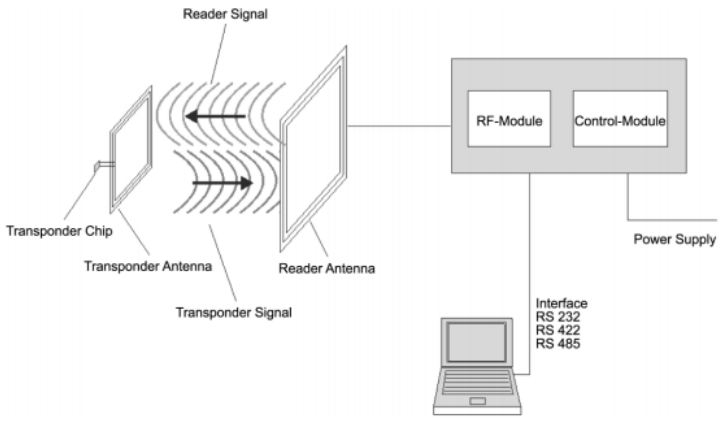


Obrázek 1: *Pasivní RFID tag (Dostupné z: http://endtimestruth.com/wp-content/uploads/2014/01/RFID-chip-and-antenna-3.png)*

- **Anténa** udává velikost tagu. Pracuje na principu indukčnosti, kde se pomocí magnetoelektrických vln indukuje střídavý proud. Tento proud nabíjí kondenzátor na minimální potřebnou hodnotu pro přenos kódu.
- **Čip** bývá malých rozměrů (cca 1mm). Uchovává obvod na zpracování signálu a také paměť RFID.

Princip komunikace

Čtecí zařízení prostřednictvím antény vysílá periodicky na svém nosném kmitočtu elektromagnetickou vlnu (rádiovou vlnu) do okolí. Objeví-li se ve vhodné vzdálenosti od antény tag, který je naladěn na stejnou frekvenci, je tato vlna přijata anténou tagu. Indukované napětí na anténě tagu, vyvolá střídavý elektrický proud, který je usměrněn a nabíjí kondenzátor v tagu. Uložená energie je použita pro napájení logických a rádiových obvodů tagu. Když napětí na kondenzátoru dosáhne minimální potřebné úrovně, spustí řídící obvody uvnitř tagu a ten začne odesílat odpověď čtecímu zařízení. Vysílání tagu je realizováno zpravidla pomocí dvoustavové ASK (Amplitude Shifting Key) modulace, která je realizována změnou zakončovací impedance antény transpondéru (anténa je buď přizpůsobena, nebo zakončena nakrátko). Modulace představuje důkladné ovlivňování tří parametrů signálu, a to je výška, frekvence a fáze amplitudy. Pomocí modulace vlny vysílané ze čtečky lze do tagu i zapisovat (pokud to umožňuje). Analýzou těchto vln kdekoliv v dosahu čtečky můžeme zpětně zrekonstruovat zprávu přijaté vlny - demodulace. Odrazy, které vznikají změnou impedance antény, jsou detekovány čtečkou a interpretovány jako logické úrovně 1 a 0. Dostatečná energie pro nabití kondenzátoru v transpondéru a schopnost detekovat přijatou odpověď transpondéru čtečkou jsou tak hlavní hardwarové podmínky fungování RFID systému. S rostoucí vzdáleností mezi čtečkou a transpondérem postupně klesá kvalita RFID signálu. Narůst šumu v základním signálu vede až k nemožnosti úspěšné detekce přijaté zprávy.[6]



Obrázek 2: *Princip činnosti pasivního RFID(dostupné z: http://www.emeraldinsight.com/doi/abs/10.1108/02640470410552947?journalCode=el)*

Rozdělení RFID systému podle frekvenčního pásma

Systém RFID využívá radiových vln, které podle typu vlnové délky procházejí různými materiály. Podle typu kmitočtového pásma lze dosáhnout na čtečku i přes nepříznivé podmínky okolního prostředí. Ovšem závisí na volbě frekvence, dle které lze rozdělit RFID systém na čtyři frekvenční pásma.[6],[11],[1]

1. **Low Frequency** (Nízké frekvence) – frekvence 125 – 134 kHZ, dosah čtecí vzdálenosti cca. 20 cm a přenosová rychlost je nízká. Nejčastěji se používají pasivní tagy s nepřepisovatelnou pamětí.
2. **High Frequency** (Vysoká frekvence) –frekvence 13,56 MHz, dosah čtecí vzdálenosti cca. 1 m a má dostatečnou přenosovou rychlost. Spolehlivé v prostorách obsahující kapaliny nebo kovy. Používají se především pasivní tágy s pamětí buď RO (Read Only – pouze čtení) nebo RW (Read Erite – možnost zápisu).
3. **Ultra High Frequency** (Velmi vysoká frekvece) – frekvence 860 – 960 MHz, přenos informací zvládá do 3 m a přenosová rychlost je vysoká. V současné době nejpoužívanější RFID pásmo využívající standart ISO 18000 určený pro knihovní systémy, docházkové systémy atd. Různá frekvenční pásma ve světě.
4. **Microwave** (Mikrovlnná frekvence) – frekvence 2,54 GHz nebo 5,8 GHz, přenos zvládá oproti ostatním na desítky metrů. Vyznačuje se vysoko přenosovou rychlostí ale s velmi špatným výkonem v přítomnosti kovu. Pro toto pásmo se nejvíce uplatňují aktivní tagy.

Platné standardy pro radiofrekvenční identifikaci

ISO 7816 Standard pro kontaktní čipové karty.
ISO 7816-1 Standard popisuje elektrické a mechanické vlastnosti karet.
ISO 7816-2 Standard popisuje velikost, pořadí, umístění a funkčnost kontaktních oblastí karty.
ISO 14443 Standard pro bezkontaktní karty pracující na frekvenci 13,56 MHz do 15 cm.
ISO 15693 Standard pro bezkontaktní karty pracující na frekvenci 13,56 MHz od 1m do 1,5m.
ISO 18000 Standard pro použití RFID v letectví.
ISO 18000-1 Standard popisuje obecné parametry RFID
ISO 18000-2 Standard popisuje parametry pro rozhraní <135kHz.
ISO 18000-3 Standard popisuje parametry pro rozhraní 13,56 MHz.
ISO 18000-4 Standard popisuje parametry pro rozhraní 2,54 GHz.
ISO 18000-5 Standard popisuje parametry pro rozhraní 5,8 GHz.
ISO 18000-6 Standard popisuje parametry pro rozhraní 860 až 930 MHz.
ISO 18000-7 Standard popisuje parametry pro rozhraní 433 MHz (ve vývoji).
ISO 11784 Standard pro RFID identifikaci zvířat. Popisuje strukturu kódu v tagu.
ISO 11785 Standard pro RFID identifikaci zvířat. Popisuje přenosový protokol. [6]

2.1.2 Více rozměrné tágy

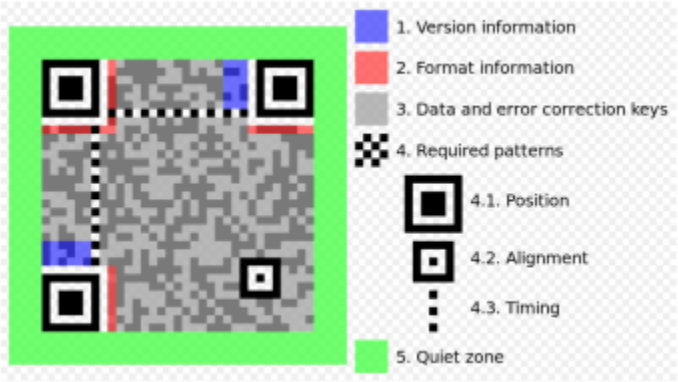
Také nazývané Bumpy Barcode. Jde o typické čárové kódy, pouze se liší ve způsobu tisku a snímání. Jsou totiž tištěny hloubkově neboli embosovány, což můžeme např. vidět u platebních karet do bankomatu. Snímání se pak provádí podle výškových změn a rozdílů.[13] V nemocničním prostředí jsou takto označeny ku příkladu chirurgické nástroje, kde je kód vyryt do materiálu, tak aby byl odolný vůči sterilizaci.

2.1.3 QR kód

Dalším velmi používaným tagem, je QR Code (Quick Response Code). Ten byl vytvořen japonskou společností Denso-Wave v roce 1994. QR kódy jsou striktně dané podle normy ISO 18004:2006.[14]

Oproti jednodimenzionálních tagu, jsou dvoudimenzionální rychlejší ve čtení a obsahují mnohem více volné paměti pro uchování dat.

Každý QR kód je tvořen čtverci, tzv. Finders, které se nacházejí v rozích kódu. Dále mezi Finders jsou řádky a sloupce vyhrazené na tzv. Timing vzory, což je střídání černých a bílých bodů. V neposlední řadě je na QR kódu vidět tzv. Aligmments, což jsou čtverce menšího tvaru než Finders, lišící se od nich černou tečkou ve čtverci.[15],[1]



Obrázek 3: *Ukázka QR kódu [1]*

Popis k obrázku číslo 3:

1. Informace o verzi. - Je to informace o použité velikosti.
2. Informace o formátu. - Podle rozpoznaného textu pozná, zda se jedná o webovou stránku nebo o telefonní číslo.
3. Data a samo opravné kódy.
4. Povinné vzory. - Jsou terče pro správné zaměření.
5. Tichá zóna. - Je ohraničení celého kódu bílou barvou. Stejně, jako povinné vzory slouží ke správnému zaměření.

2.1.4 EAN

Zkratka EAN je odvozena z anglického jazyka European Article Number. Je to jednorozměrný kód, který funguje na principu kontrastu bílé a černé čáry. Jedna čára reprezentuje jeden paměťový bit.

Na začátku vzniku čárových kódu byl obchodník, kterému vadilo zdlouhavé odbavování zákazníku u pokladen. Tohoto problému se ujmul 30 letý inženýr Norman Joseph Woodland. Za vznikem EAN kódu stojí Morseova abeceda. Tečky a čárky tehdy inženýr zapisoval na pláži do písku. Malou chybou se z teček staly čárky, které měli podobu nám tak známého čárového kódu. První ostré nasazení čárového kódu se událo v Americe 26. 6. 1974 kde, poprvé „píply“ žvýkačky za 67 centů. [10]

V dnešní době je nejpoužívanější čárový kód EAN-13. U EAN-13 jsou jednotlivé symboly zakódovány do 13-ti čísel, která jsou rozdělena na čtyři části:

- První část – první dvě nebo tři číslice identifikují zemi zaregistrovaného výrobce, pokud ale vznikl z knižní podoby ISBN, obsahuje první tři číslice čísla 978 nebo 979. Pokud vznikl z podoby časopisu ISSN, tak jsou první tři čísla 977.
- Druhá část – je kód výrobce, v závislosti na systému kódování se skládají ze čtyř nebo pěti číslic.
- Třetí část – je kód výrobku. Ten se skládá z pěti číslic.
- Čtvrtá část – je kontrolní část, která se odpočítá z modulu 10, tzv. samo detekující kód.
    - Sečtu číslice (od konce) na lichých pozicích (4+2+4+2+0+5)=17
    - Přičtu součet číslic na sudých pozicích (od konce) vynásobený třemi ((5+3+1+3+1+9)*3=36)
    - Tento součet zaokrouhlím na desítky nahoru (17+36=53) ⇒ 60
    - Kontrolní číslici získám odečtením 60-53 = 7(příklad viz. obr. 4) [11]

Obrázek 4: *Jednorozměrný EAN-13 kód. (Převzato z [11])*

3 Android

Projekt byl navržen na platformě Android. Hlavní důvody užití této platformy byla dostupnost. Android je, volně dostupná, velice rozšířena mobilní platforma, která umožňuje vývojářům, například prostřednictvím Android Studia, vytvářet mobilní aplikace. Android běží na linuxovém jádře. Programuje se v jazyce JAVA. Grafické objekty se vytváří v xml souborech.

3.1 Historie Android

Android pochází z dílny společnosti Android, Inc. Založené v roce 2003 v Palo Alto v Kalifornii v USA. Mezi jeho zakladatele se řadí Andym Rubinem, Richem Minerem, Nickem Searsem a Chrisem Whitem. V roce 2005 byla společnost odkoupena gigantem Google, Inc. V Googlu pod vedením Andyho Rubina byla vyvinuta volně dostupná (Open Source) mobilní platforma založená na Linuxovém jádře, která byla rozšiřitelná na mobilní telefony z různými hardwarovými parametry. 5. listopadu 2007 bylo vytvořeno uskupení Open Handset Alliance. Toto uskupení mělo brát Android jako výchozí platformu, která by se měla uživateli představit ve většině případech. Android byl tímto povýšen, na jaký si standart, pro mobilní zařízení. Díky své Open Source licenci za nedlouho vyšel i Android SDK pro vývojáře. Do České republiky Android zavítal prvně v roce 2009. Dnes je jedním z nejpoužívanějších platforem pro mobilní zařízení s podílem 84%. Jako druhá platforma se uvádí iOS s 12% pro produkty firmy Aple. V roce 2014 odkoupila společnost Windows známy mobilní gigant Nokia a chce tímto konkurovat na trhu mobilních zařízení s platformou Windows phone. Windows phone poměrně nová platforma zabírá na trhu pouhé 3%.

Android se postupem času vyvíjel a doplňoval funkcionalitu s technologickým postupem a zvyšováním nároku uživatele. V dnešní době je aktuální verze androidu 5.0 Lollipop (k datu 27. 12. 2014). Jednotlivé verze systému dostali název podle zákusků, jdoucích podle abecedy (Cupcake, Donut, Eclair, Froyo, Gingerbread, Honeycomb, Ice Cream Sandwich, Jelly Bean, KitKat a Lollipop). Na práci byl použit android 4.4 Kit Kat.

3.2 Struktura androidu

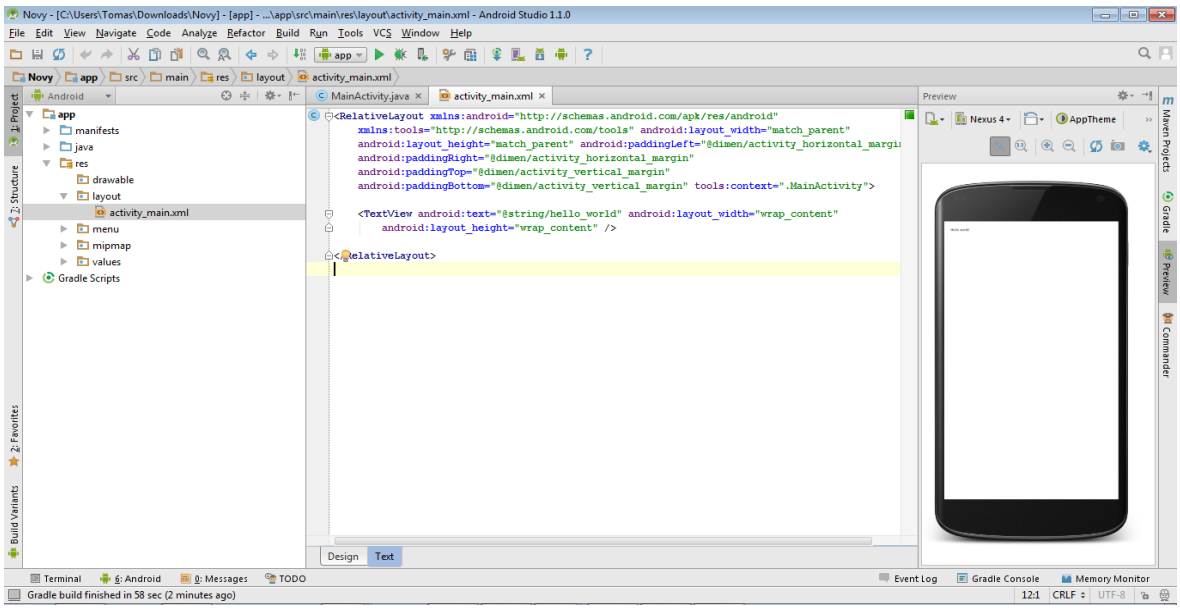
Pěti vrstvá struktura androidu, kde každá vrstva provádí odlišné operace, avšak s ostatními vrstvami spolupracuje. Tudíž se nedá tvrdit, že jednotlivé vrstvy pracují nezávazně na sobě. Jednotlivé vrstvy androidu:

- Linux Kernel (linuxové jádro)
- Libraries (knihovna)
- Android Run Time (běhové prostředí Androidu)
- Aplication Framework (aplikační rámec)
- Application Layer (aplikační vrstva)

3.3 Android Studio

Tato práce byla tvořena pomocí Android Studia verze 1.1.0. Jedná se o volně dostupné vývojové prostředí pro platformu Android. Vyvinula jej společnost Google společně s JetBrains. Android studio staví nad Community verzí prostředí IntelliJ IDEA. Díky tomu získává všechny možnosti práce s kódem (navigace v kódu, našeptávání, refaktoring, analýza kódu…). [16]

3.3.1 Uživatelské rozhraní



Obrázek 5: *Přehled uživatelského prostředí po vytvoření nového projektu (orientačně).*

Z osobního pohledu je android studio velmi přizpůsobivé uživateli například rozvržení velikosti oken a volbou široké škály podhledu. Mezi takové tři základní pohledy oken, které se objeví po založení nového projektu jsou:

- Projektové okno (Struktura projektu možnosti zobrazení: Android, Project, Packages)
- Okno pro psaní kódu (Zobrazení jednotlivých souborů JAVA nebo xml soubory)
- Previwe (online visuální podpora programátora)

Kořenová struktura

Závisí na zvolené struktuře zobrazení projektového okna. Nicméně se po spuštění projektu vygenerují pomocná zdrojová data. Ve struktuře Android a složce app se generují tyto složky:

- **Manifest** – Soubor AndroidManifest.xml je základem celé aplikace. Zde se uvádí všechny potřebné informace o deklaraci a vzájemném propojení komponent, jako např. aktivit, služeb, přijímačů či kamery nebo NFC adaptérů.
- **Java** – Pod složkou java jsou uloženy soubory psané v jazyce JAVA.
- **Res** – Složka slouží k uchování všech externích prostředků, jež jsou v Android aplikaci, např. GUI (Graficko uživatelské rozhraní), textové informace, různé druhy souboru.

3.3.2 Aplikace v Android Studiu

Aplikace vyvíjena pro OS Android prostřednictvím Android Studia se může dělit na tři typy z hlediska uživatelské práce a to na:

Aplikace na popředí

Jedná se o aplikace, které jsou užitečné pouze v případě, když běží v popředí a jejich činnost je efektivně pozastavena, když přejde na pozadí. Typickým příkladem jsou hry nebo aplikace pracující s mapami.[7]

Aplikace na pozadí

Hlavní činnost těchto aplikací probíhá mimo aktuální běh zařízení. Pracují tzv. na pozadí. Příkladem mohou být aplikace sledující telefonní hovory nebo SMS.[7]

Aplikace s přerušovanou činností

Tyto aplikace také většinou pracují na pozadí zařízení, avšak očekává se od nich komunikace s uživatelem formou upozornění na určité události. Typickým příkladem může být přehrávač médií.[7]

Vývojář si musí při vyvíjení aplikací pečlivě zvolit, na co bude aplikace a jak bude aplikace pracovat. Poté aplikaci přizpůsobit a to i ze stránky dizajnu. Když vývojář vyvíjí aplikaci, potřebuje komponenty pro stavbu aplikace. Komponenty mohou být aktivity, záměry, služby atd.. Komponentu si lze představit, jako prezentační vrstvu aplikace.

Activity (aktivity)

Aktivita poskytuje dialogové okno, prostřednictvím kterého se může komunikovat. V aplikaci se určuje jedna aktivita jako hlavní, která se zobrazí ihned po spuštění aplikace. Dále může pokračovat dalšími aktivitami, které se spouštějí za účelem nějaké funkce. Vždy když se spustí další aktivita, pozastaví se systém předchozí aktivity a ta je uložena do zásobníku. Zásobník funguje jako systém LIFO (last in, first out – poslední dovnitř, první ven). Mechanismy pro zachování zdrojů jsou nazývány životní cyklus aktivity.[7]

Životní cyklus aktivity se začíná metodou **onCreate**, tato metoda se volá, když se aktivita vytvoří. Viditelnou pro uživatele se stává po zavolání metody **onStart**. Pokud není třeba uživateli ukazovat, co se v aktivitě dělá, volá se metoda **onStop**. Pokud v aplikaci nastane přerušení např. příchozí hovor, vyvolá se metoda **onPause**. Po skončení hovoru se vrací do aktivity pomocí metody **onResume**. Aktivitu lze ukončit z vnitřku pomoci metody finisch(), ta volá metodu **onDestory**, která aktivitu zničí. Dále se volá metoda **onRestart**, pokud je aktivita zastavena a je třeba jí obnovit.

Vytvořením nové aktivity je nutností uvést aktivitu do souboru AndroidManifest.xml. Po vytvoření aktivity v Android Studiu se vytvoří grafický layout (xml soubor obsahující grafické elementy aktivity) a předpřipravený java soubor.

4 Uložiště

V této práci jsou data ukládána na externí server uložený na VŠB pod adresou https://rfidexpert.vsb.cz. V praxi ale komunikuji se serverem přes PHP skripty. Taková komunikace se vyznačuje vyšší bezpečností. Pro android z verzí vyšší API úrovně než 4. 2 je přímé dotazování SQL příkazy zakázáno (možnost zneužití dat). Zejména, když se jedná o pacientská osobní data, kde je nutno dodržovat zejména zákon č. 20/1966 Sb., *O péči a zdraví lidu, ve znění pozdějších předpisů, zákon č. 256/1992 Sb., O ochraně osobních údajů v informačních systémech a metodický návod ministerstva INF-11341, k zabezpečení a ochraně osobních údajů v IS provozovaných ve zdravotnických zařízeních.*[4]

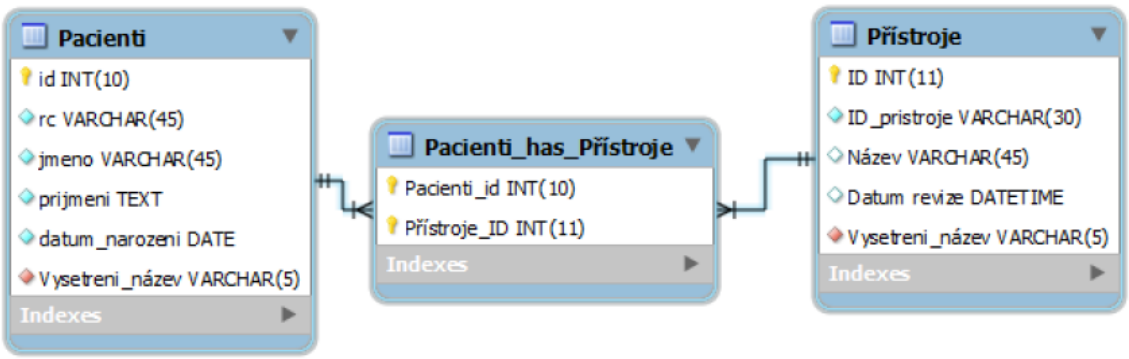
4.1 Databáze MySQL

MySQL je databázový systém vytvořený švédskou firmou MySQL AB, nyní vlastněný společností Sun Microsystems, dceřinou společností je Oracle Corporation. Je dostupný jako open source, ale také pod placenou licencí. Považuje se za nejpoužívanější relační databázový systém, který vyčnívá svou stabilitou a poměrně slušnou rychlostí. Databáze MySQL je jeden z prvních hojně rozšířených systémů. Práce s tímto systémem se dá využít v C, C++, Java, Perl, PHP, Python, Tcl, Visual Basic nebo. NET.Momentální nejnovější verze MySQL, kterou Oracle nabízí je 6. 5. 11. V bakalářské práci byla použita verze Workbench 5. 6. 24 CE. [6]

Tato diplomová práce navazuje na práci bakalářskou, kde byly databázové systémy uvedené v širší podobě. V této práci uvádím jen základy pro zvládnutí informačního přehledu o databázi.

4.1.1 Relační databáze

V této práci byl použit relační datový model. Model má jednoduchou strukturu, data jsou organizována v dvourozměrných tabulkách, které se skládají z řádků a sloupců. V těchto tabulkách jsou prováděny všechny databázové operace. Velkou předností relačního modelu je schopnost tabulky spojovat a vytvářet další dvourozměrné tabulky.[5]



Obrázek 6: *Relační model dat.*

Definice pojmů v relační databázi

Typ entity

Je známo, že relační databáze shromažďuje data do tabulek. Tabulky jsou sestaveny tak, aby setřiďovaly podle charakteru dat. To znamená, že tabulka s typem entity pacient bude shromažďovat záznamy všech pacientů.

Entita

Entita je osoba, místo, věc, událost nebo myšlenka, o níž shromažďujeme nějaká data. Jinými slovy entity jsou „předměty" z reálného světa, které jsou pro nás dostatečně zajímavé, že o nich sledujeme údaje a zaznamenáváme je do databáze. [17]

V databázi přesněji pak v typech entit (tabulkách) je entita reprezentovaná řádkem. Například máme evidovat pacienty. Každý záznam pacienta je entita.

Atribut

Atribut je vlastnost entity. Popisuje důležité informace entity tak, aby možno co nejvíc přiblížil problematiku reálného světa. Čím více bude entita popsána, tím více budeme shromažďovat dat a systém bude chytřejší na úkor množství evidovaných dat.

V databázi přesněji pak v typech entit (tabulkách) je atribut reprezentovaný sloupcem. **Složený atribut** – skupina atributů, která má společný význam nebo použití. Typickým příkladem je atribut adresa, který je složen z jednoduchých atributů PSČ, město, ulice, číslo domu.

Doména

Množina přípustných hodnot přiřazená jednomu nebo více atributů. Například doména všech hodnot čísla pracovníka.

Relace

Relace jsou informační mosty tabulek. Relací propojíme tabulky a tím vytvoříme nad tabulku, která bude rozšířena o informace právě propojených tabulek. Např. tabulku pacienti, kterou propojíme s tabulkou přístroje. Dovedeme si tak vyhledat, jaký přístroj byl použit na pacienta.

Klíče

Existují speciální druhy atributů, které umí jednoznačně určit pozici nebo entitu v tabulce.

- Superklíč - Atribut, nebo množina atributů, které jednoznačně identifikují entitu v relaci.
- *Kandidátní klíč* - Superklíč, který obsahuje minimální počet atributů potřebných k jednoznačné identifikaci entity. Je zapotřebí vždy definovat jako NOT NULL.
- *Primární klíč* - Jeden vybraný kandidátní klíč
- *Alternativní klíč* - Zbývající kandidátní klíče.
- *Cizí klíč* - Představuje odkaz mezi tabulkami. Nejčastěji to bývá primární klíč vázané tabulky relací.

4.2 Interní databáze SQLite

V této práci byla použita i databáze SQLite, která reprezentuje standardně relační databázový systém, jenž je obsažen v jedné malé knihovně. Jedná se o systém, který je velmi šetrný k paměti zařízení a poměrně rychlý vůči robustnějším databázím, na úkor oříznutí standartních funkcí jako např.:

- Absence cizího klíče
- Absence datového typu DATE
- Žádná kontrola datových typů (Sting, Intiger, Double…)
- Omezený ALTER TABLE

Tento systém není určen primárně pro Androidy, ale může být používán například v těchto programovacích jazycích C, C++, Java, Delphi, PHP, Perl, Python a TCL.

Databázový systém SQLite je klasická databáze využívající dialekt jazyka SQL. Tento dialekt je postaven na standartu SQL-92.[7]

4.2.1 Vytvoření a aktualizace s třídou SQLiteOpenHelper

Chcete-li vytvořit a aktualizovat databázi ve vašem Android aplikace můžete vytvořit podtřídu SQLiteOpenHelper třídy. V konstruktoru vaší podtřídy voláte super () metodu SQLiteOpenHelper , určující název databáze a aktuální verzi databáze. V této třídě je potřeba připsat následujících metodu k vytvoření a aktualizaci databáze. [19]

- OnCreate () - se nazývá rámce, v případě, že databáze je přístupná, ale ještě není vytvořen.
- ONUPGRADE () - volal, pokud je verze databáze se zvyšuje v kódu aplikace. Tato metoda umožňuje aktualizovat existující schéma databáze, nebo k poklesu stávající databáze a znovu ji přes OnCreate () metodu.

V práci je tvorba databáze SQLite popsána v kapitole 7.2.1.

5 Popis projektu

Projekt byl navrhnut na popud zvyšování kvality péče ve fakultní nemocnici Ostrava (FNO). Práce se zaměřuje na digitalizaci záznamu intenzivní péče (eZIP). Jedná se o sběr dat, které se vztahují k péči o pacienta. Data se sbírají pomocí mobilního zařízení a technologie internet of Things (internet věcí). Tyto data tvoří informace o postupu zdravotnické péče na urgentním příjmu.

Práce vychází z analýzy tvořené v bakalářské práci. [2] Projekt je tvořen v Android Studiu verze 1.1.0. Aplikace je tvořena na mobilním telefonu Samsung S4 a operačním systémem Android 4.4.2. data jsou uložena na linuxovém serveru dostupném na adrese rfidexpert.vsb.cz. Databáze je spravována pomocí MySQL Workbench 6.3 CE.

5.1 Stávající řešení

Dosavadní záznam intenzivní péče (ZIP) je vedený papírovou formou. Každý pacient je registrován v nemocničním informačním systému (NIS) a je mu přiděleno identifikační číslo. Štítek s identifikačním údaji, jako jméno, příjmení, datum narození, bydliště, je vytištěn a nalepen na dokument ZIP. Dokument je poté umístěn u pacienta, tak, aby se veškeré úkony mohli do dokumentu zaznamenávat bezprostředně při, nebo po ošetření. ZIP je vedený do jednotlivých části popsaných v kapitole 5.1.2. a k nahlédnutí v příloze B, C. Záznam vyplňuje vždy evidenční zdravotník, který se do dokumentu podepisuje. Kontrolu provádí i lékař, který zkontroluje, zda záznam souhlasí a stvrzuje taktéž svým podpisem. ZIP se píše vždy přes kopírovací papír, tak aby byla k dispozici kopie záznamu. Originál je uchován na oddělení urgentního příjmu a kopie putuje s pacientem na jiné oddělení. Diagram stávajícího řešení je k nahlédnutí v příloze E.

5.1.1 Příjem pacienta

Na urgentní příjem fakultní nemocnice Ostrava se pacient může dostat několika způsoby. Nejčastěji to bývá zdravotní záchrannou službou (ZZS). Ale může to být leteckou záchrannou službou (LZS), nebo vlastní dopravou. Velmi zřídka je pak pacient přeposlán z interního příjmu na příjem urgentní.

- Způsoby pacientova přijetí:
- Záchranná zdravotnická služba (ZZS)
- Letecká zdravotnická služba (LZS)
- Vlastní doprava
- Z jiného oddělení

Pokud pacient dorazí ZZS nebo LZS jsou to ohlášené případy. Evidenční sestra dostane zprávu o přijíždějícím pacientovi. Dozvídají se stav pacienta a jeho počáteční diagnózu. Zahájí přípravu na pacienta. Nachystá se dokumentace a potřebné zdravotnické pomůcky na ošetření. Uvedeno na

příkladu. Na centrále urgentního příjmu dostávají záznam o převozu pacienta. Zpráva se předá týmu, který začne s přípravou manipulačního lůžka. Evidenční sestra se stará o interní dokumentaci tzn., napíše číslo pacienta, nachystá záznam intenzivní péče (ZIP), spouští IS, dale podle předběžné diagnózy svolává odborné konzilia a informují se sály plus CT vyšetřovny. Evidenční sestra pak informuje o poloze pacienta, jelikož otevírá garáž (místo příjezdu ZZS), nebo dveře k heliportu (LZS). Tyto místa jsou monitorovány videotechnikou. Velkou výhodou těchto převozů je včasné varování a komunikace se zdravotní službou. [2]

Registrace pacienta do nemocničního informačního systému se provádí na centrálním počítači. Existují dva systémy pro registraci pacienta do NIS, CLINICOM a CARE CENTRUM. Registrujeme pomocí dokladu. Pokud pacient u sebe žádné doklady nemá, komunikujeme s pacientem. V případě, že je pacient v bezvědomí nebo z jiného důvodu nekomunikuje, vedeme pacienta do systému jako XY a místo jména píšeme místo, odkud byl pacient převezen, dále pohlaví, datum a diagnózu. Tak aby byl pacient zpětně dohledatelný.

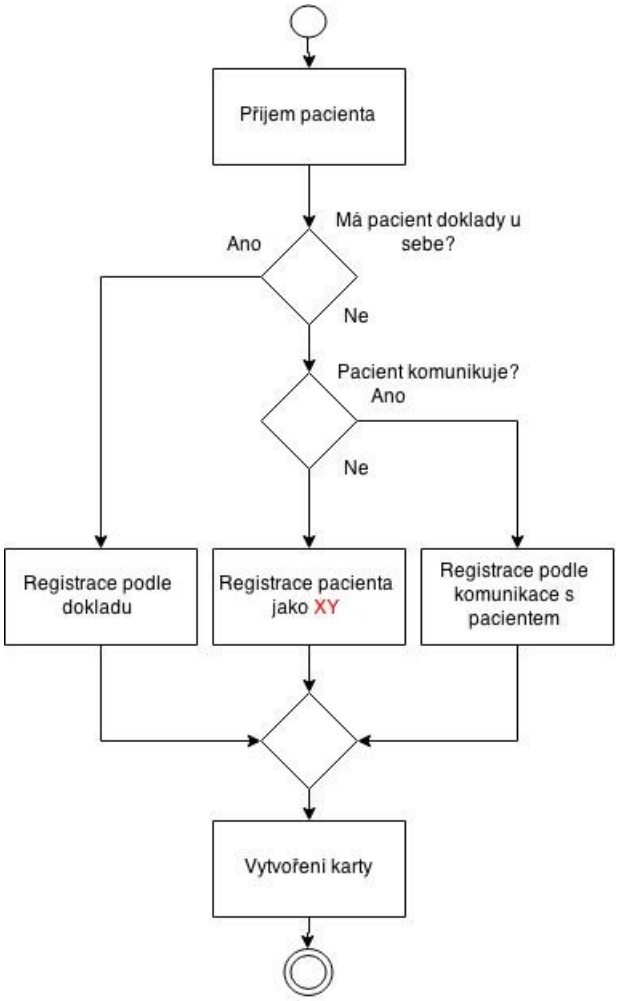


Diagram č. 1: *Diagram aktivity příjmu pacienta.*

5.1.2 Záznam intenzivní péče

Záznamu intenzivní péče (ZIP) je interní záležitost FNO. Způsob papírové evidence práce na pacientovi. Formulář velikosti A3 přehledně strukturalizován. Do formuláře se zapisují smluvenými symboly, podle VŘA o vypisování dokumentace (vydala FNO), v časové linii např. léky, vitální funkce tzn. krevní tlak, puls ale také výkony, laboratorní vzorky, krevní banka, konzilia, výška, váha, bolest a mnoho dalších parametrů, které jsou později uvedeny. Tento záznam se přikládá ke kartě pacienta a putuje s ním na další oddělení. Pro každého pacienta je jeden záznam intenzivní péče.

Mezi výhody ZIP řadíme:

- Přehledný uspořádaný arch A3
- Intuitivní vyplňování (k dispozici vzor na oddělení)
- Rychlá odezva čtení, zápis
- Je označen identifikačním štítkem pacienta se zákl. údaji: jméno, příjmení, rodné číslo, pojišťovna, bydliště, pracovní diagnóza se kterou byl přijat - ne konečná
- Obsahuje podpis evidenční sestry plus podpis ošetřujícího lékaře.
- Přesný záznam gramáže medikamentů v čase
- Používání pouze schválených zkratek ve FNO
- Produkují se vždy dva záznamy originál a kopie. Originál se nechává podepsat u převozu pacienta na jiné oddělení sestrou, která pacienta přebírá a uchovává se v kartě na urgentním příjmu. Kopie se předává přebírající sestře s kartou pacienta.

Mezi nevýhody ZIP řadíme:

- Velikost dokumentu (zabírá pracovní plochu)
- 2 listy se mohou posunout a tím pádem snadno zanést chybu v časové ose, ale také v záznamu fyziologických funkcí – tlak, puls – jiné hodnoty na kopii
- V originále jsou záznamy i barevně – opiáty, antibiotika, transfuzní přípravky, kopie je jednobarevná
- Riziko potřísnění, vodou, dezinfekcí
- Nečitelný rukopis a tím pádem záznam některých pracovníků
- V případě akutního stavu nepřesný záznam do papírové podoby – kolonky jsou po 10 minut
- Není záznam EKG, v čase, značí se pouze hodnota pulsu
- V případě nouze vypisování zpětně
- Jediný přístup k záznamu
- Časový záznam je spíše orientační

Struktura ZIP

Dokument je vedený do jednotlivých částí. Výsledná škála informací je důležitá ke správné digitalizaci záznamu. Dokument lze rozdělit do jednotlivých tříd:

- **Hlavička pacienta –** Zde jsou vypsány informace o pacientovi: jméno, příjmení, výška, váha, alergie, osobní věci, časy a datum, identifikační štítek, místo předání.
- **Životní funkce –** Tyto informace jsou snímány z monitoringu v intervalech 5 minut. Evidují se krevní tlaky (systolický, diastolický) a krevní puls.
- **Léky –** Zapisují se infuze s dobou trvání podání léku. Dále perfuzor s gramáží podaného léku a další způsoby podaného léku pacientovi. Vše se vypisuje do časové osy.
- **Ostatní –** V této třídě se zapisují ostatní informace, které mají zapotřebí obsahovat časovou stopu např.: stav vědomí, ventilace, saturace kyslíku v krvi a kyslík přiváděný do pacienta, tělesná teplota, odsávání.
- **Laboratoř –** Zde se píši vzorky, které byli poslány na rozbor (biochemie, krevní obraz, koagulace, moč)
- **Krevní banka –** Zaznamenává se, jestli mu byly odebrány vzorky pro stanovení krevní skupiny stylem ANO/NE. Dále podle situace se vypíšou žádanky o krevní transfuzi. V případě podání krve se zapisuje, o jaký krevní vak šlo.
- **Vyšetření –** V dokumentu ZIP jsou zkratky jednotlivých vyšetření, které se na pacientovi nejčastěji provádějí. Zaškrtnutím pole se zaznamená jednotlivé vyšetření.
- **Výkon –** Podobně jako vyšetření jsou k dispozici zkratky výkonu s možností poznámky.
- **Konzilia –** Zkratky jednotlivých odborností lékařů ošetřujících pacienta.
- **Doplňující vyšetření –** Zde se uvádí např. vyšetření potvrzující užití drog nebo alkoholu.
- **Poznámka –** Prostor pro doplňující informace.

5.1.3 Ukončení záznamu pacienta

Do ZIP se vypíše čas ukončení pacientova záznamu. Dále se vypíše oddělení, kde se pacient posílá. Zkontroluje se záznam a podepíše evidující sestrou a lékařem. Do záznamu je také dobré psát, kde má pacient osobní věci a v jakém stavu jsou. Dále jestli pacient má doklady u sebe nebo ne.

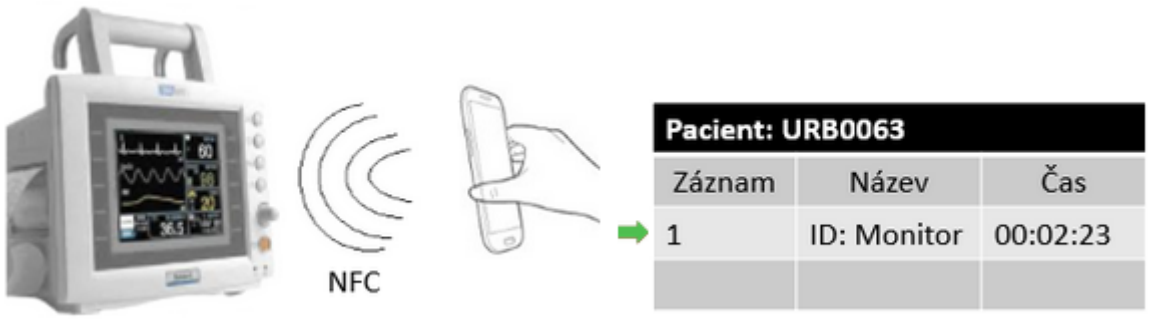
5.2 Navrhované řešení

Digitalizace záznamu nese sebou svá rizika, avšak bezesporu i klady v efektivitě práce. Záleží na návrhu elektronického systému, jakým způsobem vylepší záznam intenzivní péče. Systém musí čas šetřit a být velice intuitivní. Ovládání musí být jednoduché a přehledné. Ideálním řešením je intuitivní našeptávání systému. Co by se tedy změnilo implementací systému?

Příjem pacienta se děje od okamžiku přidělení identifikačního tágu a registrací pacienta v systému. Registrace se provádí buď pomocí mobilního zařízení, nebo pomocí uživatelské aplikace na centrálním počítači. Prvotním úkonem registrace je přidělit pacientovi jednoznačné číslo, pod kterým, bude vedena jeho dokumentace. Druhotným úkonem registrace je doplnění osobních údajů pacienta přes nemocniční informační systém nebo založení nové karty.

Práce na pacientovi se bude ukládat přečtením identifikačního čísla dané entity a přiřazení entity k profilu pacienta. Tím se vytvoří záznam o užití. Je zapotřebí přistupovat ke každé entitě podle jejího typu. Je tedy zapotřebí pracoviště osadit odpovídající technologií tak aby celkový proces byl efektivní a neomezoval péči pacienta.

Po ukončení léčby na urgentním příjmu se zpráva ukládá do databáze pacientů.



Obrázek 7: Princip evidence přístroje do eZIP.

6 Analýza

Vývoj informačního systému vyžaduje provést analýzu. V této práci byla analýza prováděna standartním způsobem. Za prvé se shromáždí požadavky kladené na systém. Poté se vytvoří datová analýza. Výstupem datové analýzy je konceptuální schéma (viz. dále). Po datové analýze se přistupuje k funkční analýze dat. Tedy popisujeme funkci systému a sledujeme pohyb a změnu dat.

Analýza prostředí urgentního příjmu FNO byla prováděna v bakalářské práci roku 2013. [2] V této práci byla doplněna na aktuální stav a z části i znázorněna.

6.1 Požadavky

Požadavky systému jsou prvotní body k vývoji informačního systému. Požadavky vždy klade zadavatel a vývojář je musí respektovat nebo po domluvě korigovat. V této práci byli zadány tyto požadavky:

- Systém musí zachovat škálu evidujících informací, případně dosavadní škálu rozšířit.
- Systém musí být čitelný podle časové osy vyšetření
- Systém musí být přehledný jednoduchý na obsluhu a rychlý
- Systém musí evidovat všechny úkony, které se provádějí na pacientovi v daném čase.

6.1.1 Požadavky na techniku

Technika musí splňovat normu ČSN IEC 930:1994 (36 4890) o zásadách bezpečného používání zdravotnických elektrických přístrojů prdo administrativní a zdravotnický personál.

6.2 Datová analýza

Datová analýza charakterizuje data, která se v databázi vyskytují. Výsledkem je konceptuální schéma, což jednoznačně definuje charakter dat. Charakter dat je důležitý v sestavení databáze, tak aby databáze byla schopna evidovat požadovaná data.

6.2.1 Specifikace dat

Každý nový pacient má svojí kartu. Každá karta může obsahovat několik záznamů. Záznam je informace o provedení akce na pacientovi. Každý takový záznam obsahuje datum a čas, dále pak ID pacienta a ID konkrétního záznamu. Konkrétní záznam vychází ze škály informací, které se zapisují do dokumentace (viz. kap. 5.1.2). Navíc záznam po načtení je z části editován. Ku příkladu léky. Sejmutím identifikačního čísla mobilním zařízením na konkrétním léku, se vytvoří záznam v čase o podání léku s konkrétním ID. Do záznamu daného léku se po sléze dodatečně dopisuje množství podaného léku.

6.2.2 Lineární zápis dat

Vhodnou metodou pro datovou analýzu je lineární zápis. Tento zápis popisuje entitu škálou evidovaných atributů. Jedná se tedy o výpis atributů, které mají své označení. Bývá zvykem, že atribut na prvním místě zápisu je podtržen a znamená hlavní klíč tabulky. Dále se vypisují atributy pro daný typ entity. V poslední řadě se kurzívou vypisují cizí klíče.

V práci byl vytvořen lineární zápis dat, který popisuje osobní data pacienta:

- **Pacienti** (id, rc, jmeno, prijmeni, kod_pojistovny, mesto, okres, ulice, cp, psc, datum_narozeni, pohlavi, vyska, vaha, alergie, zamestnani, tel_blizka_osoba, kontakt_prace, nabozenske_vyznani, cas_nacteni, cas_ukonceni, kod_pacienta)

Dále pak lineární zápis tabulek, které jsou v práci použity:

- **Leky_ZIP** (id, kod_leku, mnozstvi, nazev, zpusob_podani, podnazev, poznamku, typ)
- **Vykon_ZIP** (id, nazev, kod_vykonu, poznamka)
- **Krevni_banka_ZIP** (id, kod_konzervy, lhuta, druh, krevni_skupina, mnozstvi, poznámka)
- **Vysetreni_ZIP** (id, kod, nazev, poznamka)
- **Konzilia_ZIP** (id, kod_konzilia, nazev)
- **Laborator_ZIP** (id, nazev, kod, podnazev, poznamka)
- **Pristroje** *(*id, kod, evidencni_čislo, nazev, lokalizace)

Kardinalita vztahu těchto tabulek odpovídá N:N (např. Jeden pacient může mít N léku a jeden lék (paralen) může mít N pacientů). To vede k vytvoření tak zvaných vazebních tabulek. Tyto vazební tabulky tvoří unikátní záznamy konkrétního pacienta a konkrétního léku.

Lineární zápis vazebních tabulek:

- **Pacient_has_leky_ZIP** (id, cas_nacteni, Mnozstvi_leku, *Pacient_id*, *Leky_ZIP_id)*
- **Pacient_has_Vykon_ZIP** (id, cas_nacteni, *Pacient_id*, *Vykon_ZIP_id)*
- **Pacient_has_Konziliia_ZIP** (id, cas_nacteni, *Pacient_id*, *Konzilia_ZIP_id)*
- **Krevni_banka_ZIP_has_Pacient** (id, cas_nacteni, *Pacient_id*, *Krevni_banka_ZIP_id)*
- **Pacient_has_Vysetreni_ZIP** (id, cas_nacteni, *Pacient_id*, *Vysetreni_ZIP_id)*
- **Pacient_has_Pristroje** (id, cas_nacteni, parametry, *Pacient_id*, *Pristroje_id)*
- **Pacient_has_Osobni_vecy** (id, cas_nacteni, *Pacient_id*, *Osobni_vecy_id)*

6.2.3 Datový slovník

Datový slovník charakterizuje data atributů. Popisuje jejich typ, délku, defaultní hodnotu, unikátnost, integritní omezení a poznámky. V datové analýze projektu se zaměřujeme na osobní data pacienta.

Tabulka č. 1: Datový slovník pro typ entity Pacienti

Typ Entity	Atribut	Datový typ	Délka	Klíč	NULL	IO	Poznámka
Pacienti	id	INT	100	A	N	Při. klíč	Auto. navýšení
	rc	BIGINT	10	N	A		rodné číslo
	jmeno	VARCHAR	45	N	A		jméno
	prijmeni	VARCHAR	45	N	A		příjmení
	kod_pojistovny	INT	5	N	A		adresa
	město	VARCHAR	100	N	A		adresa
	obec	VARCHAR	100	N	A		adresa
	ulice	VARCHAR	100	N	A		adresa
	cp	INT	11	N	A		adresa
	psc	INT	5	N	A		adresa
	datum_narozeni	DATE	-	N	A		
	pohlavi	ENUM *	-	N	A		*('m', 'z', 'n')
	vyska	INT	9	N	A		
	vaha	INT	9	N	A		
	alergie	VARCHAR	100	N	A		
	zamestnani	VARCHAR	100	N	A		
	tel_blizka_osoba	INT	9	N	A		
	Kontakt_prace	INT	9	N	A		
	nabozenske_vyznani	VARCHAR	100	N	A		
	cas_nacteni	DATETIME	-	N	N		čas příjmu pacienta
	kod_pacienta	VARCHAR	100	N	A		QR, EAN, RFID

*A= ANO, N= NE.

Tabulka č. 2: Změněna vazební tabulka Pacient_has_leky_ZIP

Typ Entity	Atribut	Datový typ	Délka	Klíč	NULL	IO	Poznámka
Pacient_has_leky_ZIP	id	INT	100	A	N	Primární klíč	auto. navýšení
	cas_nacteni	DATETIME		N	N		
	Pacient_id	INT	100	A	N	Cizí klíč	
	Leky_ZIP_id	INT	100	A	N	Cizí klíč	

*A= ANO, N= NE.

6.2.4 ERD model dat

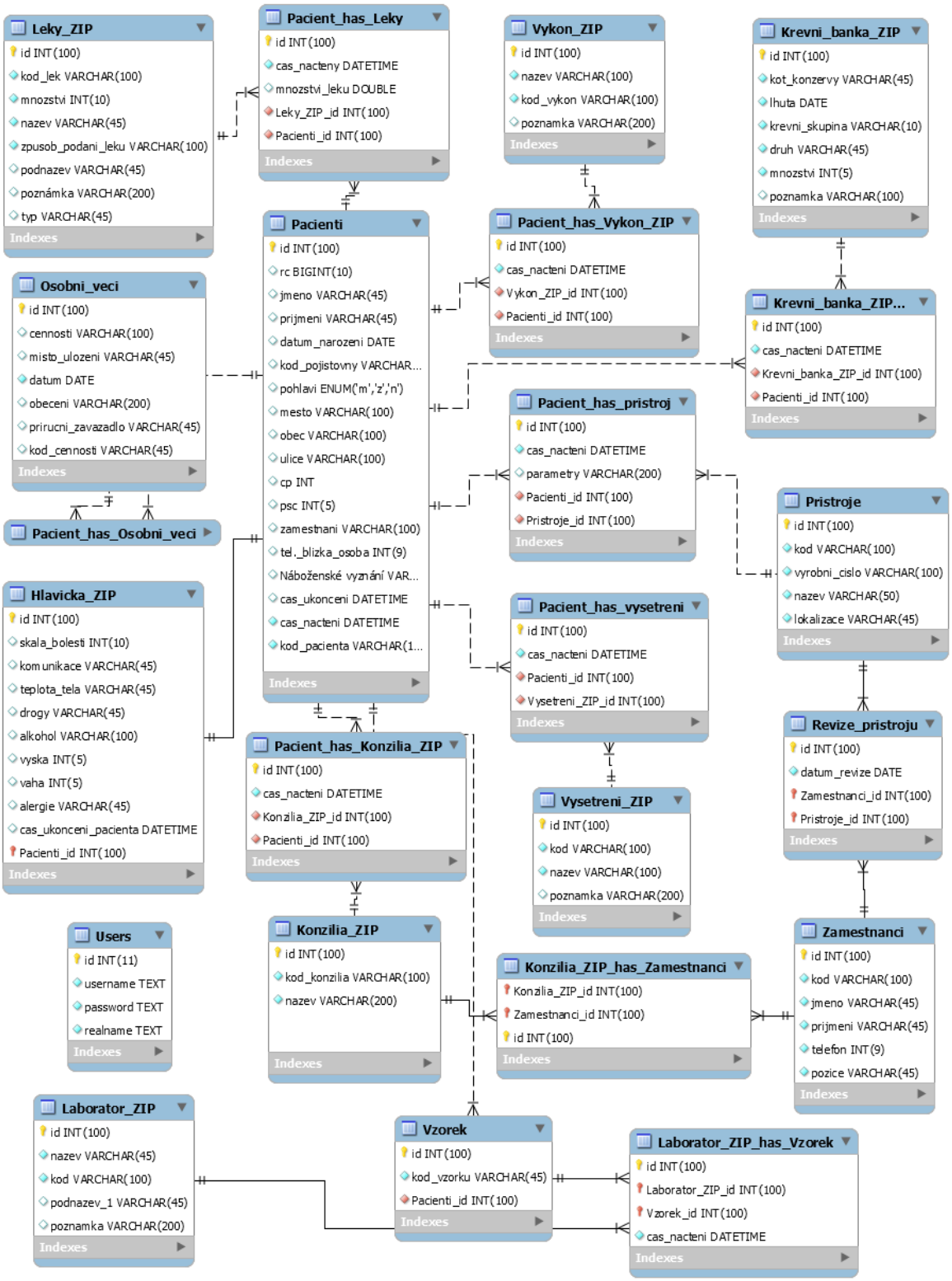


Diagram č. 2: *ER diagram. Vztahy mezi tabulky.*

6.3 Funkční analýza

Ideálním ukazatelem funkce systému je diagram datových toků. Diagram datových toků (DFD, Data Flow Diagram) patří k nejpoužívanějším nástrojům strukturované analýzy informačních systémů. Zvýrazňuje funkční vlastnosti systému. Systém zobrazuje jako síť procesů, představující místa transformace dat, zásobníků jako míst ukládání dat - kartotéky, soubory- a externích entit, které reprezentují okolní prostředí systému. Tyto spojujeme pomocí datových toků.[18]

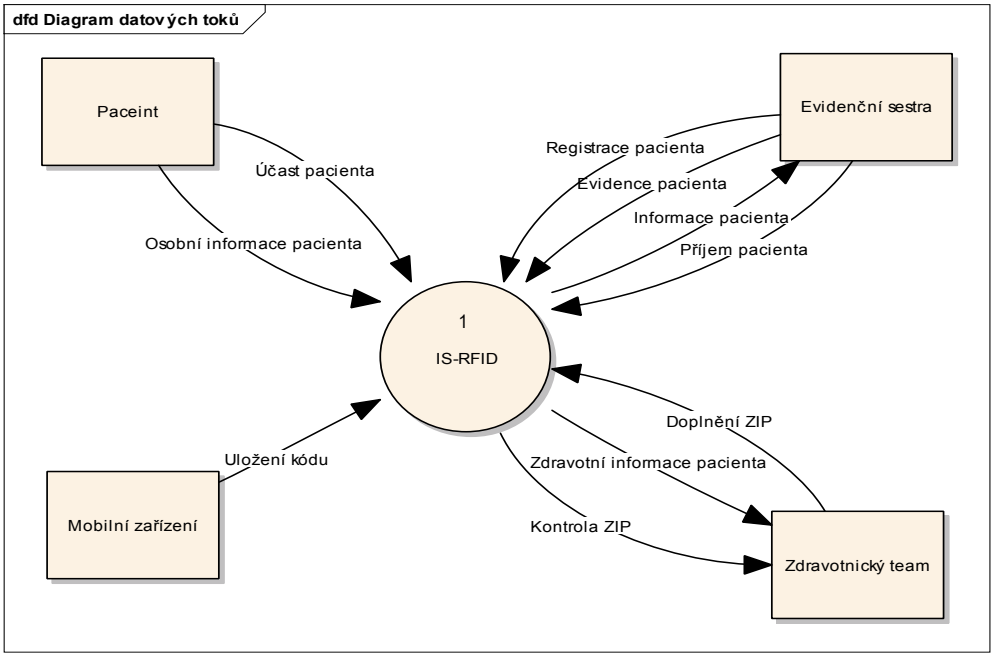


Diagram č. 3: *Diagram datových toků popisující vrchní hierarchii.[2]*

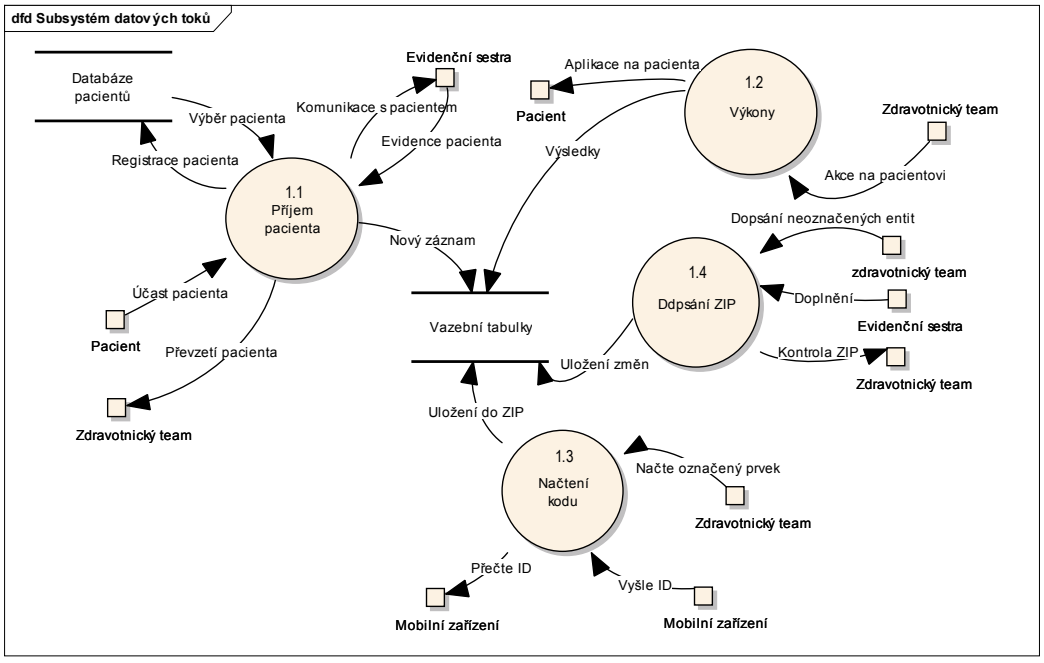


Diagram č. 4: *DFD popisující práci a užití systému v praxi.[2]*

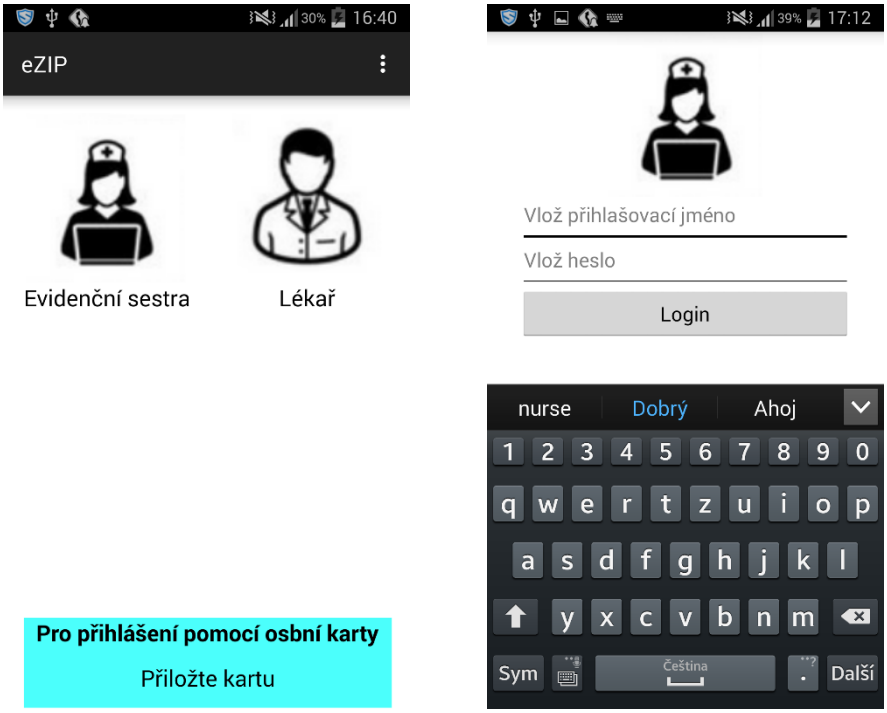
6.4 Návrh mobilní aplikace

Při návrhu mobilní aplikace je potřeba si dobře ujasnit základní funkcionalitu. Kolem této funkcionality vytvořit intuitivní rozhraní pro uživatele, tak aby bylo ovládání příjemné, nenáročné a rychlé. V práci se vychází z toho, že aplikace bude použita v nemocnici. Hlavní funkcionalitou je vytvoření záznamu intenzivní péče pacienta. Vzhledem k tomu, že se pracuje s pacientskými daty, musí aplikace být bezpečná a spolehlivá.

Pokud mobilní aplikace používá externí hardware např. kamera, NFC nebo wifi připojení je nutné tyto nároky uvést a přizpůsobit typu zařízení.

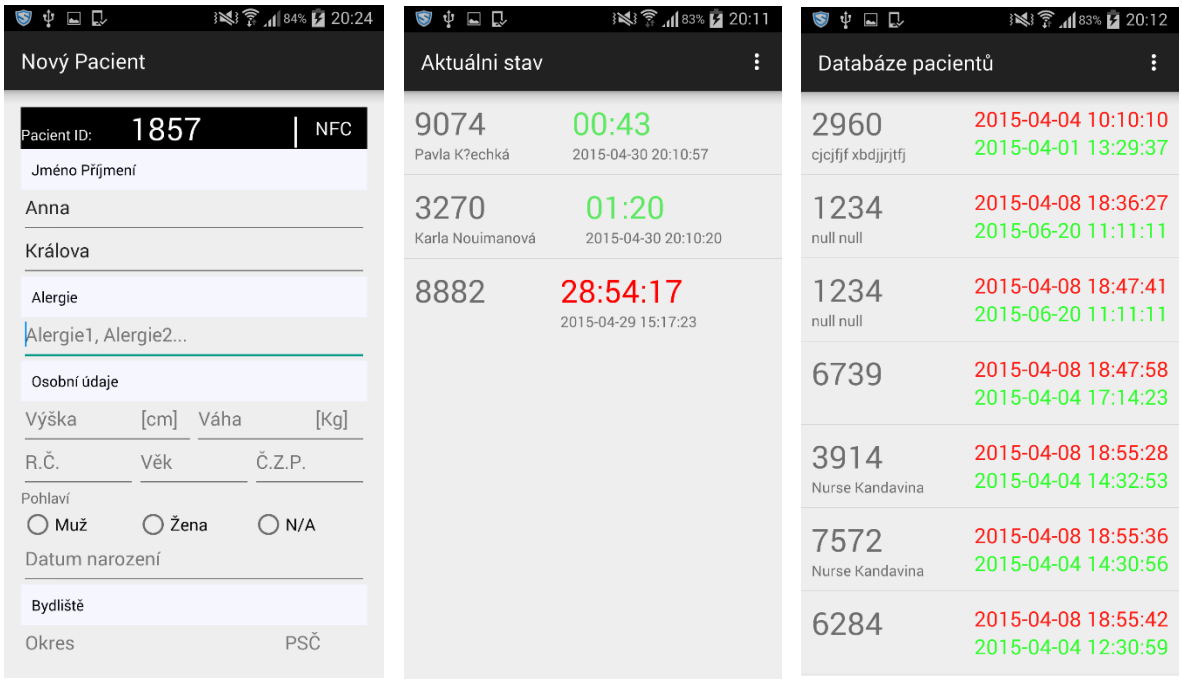
6.4.1 Návrh uživatelského prostředí

Uživatel po zapnutí aplikace je vyzván k přihlášení se do systému. Do systému je možné se přihlásit dvěma způsoby. Přihlášení jako sestra nebo lékař. Každá volba vede k přihlašovací aktivitě, která nabízí dva edit boxy a to na uživatelské jméno a uživatelské heslo. Po zadání správných údajů se uživatel přihlásil do aplikace. Po chybném přihlášení dostane oznámení o chybném přihlášení a možnost dalších pokusu. Do systému se lze přihlásit nikoli registrovat. Registraci provádí správce. Mimo jiné se uživatel může přihlásit pomocí služební karty pro rychlé přihlášení pomocí NFC (bez nutnosti zadávání přihlašovacích údajů).



Obrázek 8: *Uživatelské prostředí pro přihlášení se do systému.*

Uživatel se dostane na hlavní nabídku orientovanou do čtvercových tlačítek. Tlačítka jsou tvořená ikonami jednotlivých sekcí. Uživatel přihlášený, jako sestra, má možnosti přejít na registraci pacienta, aktuální stav, databáze pacientů a připravené tlačítko pro sledování polohy pacienta na urgentním příjmu.



Obrázek 9:Uživatelské prostředí pro registraci, aktuální a databázi pacientů.

Nový pacient

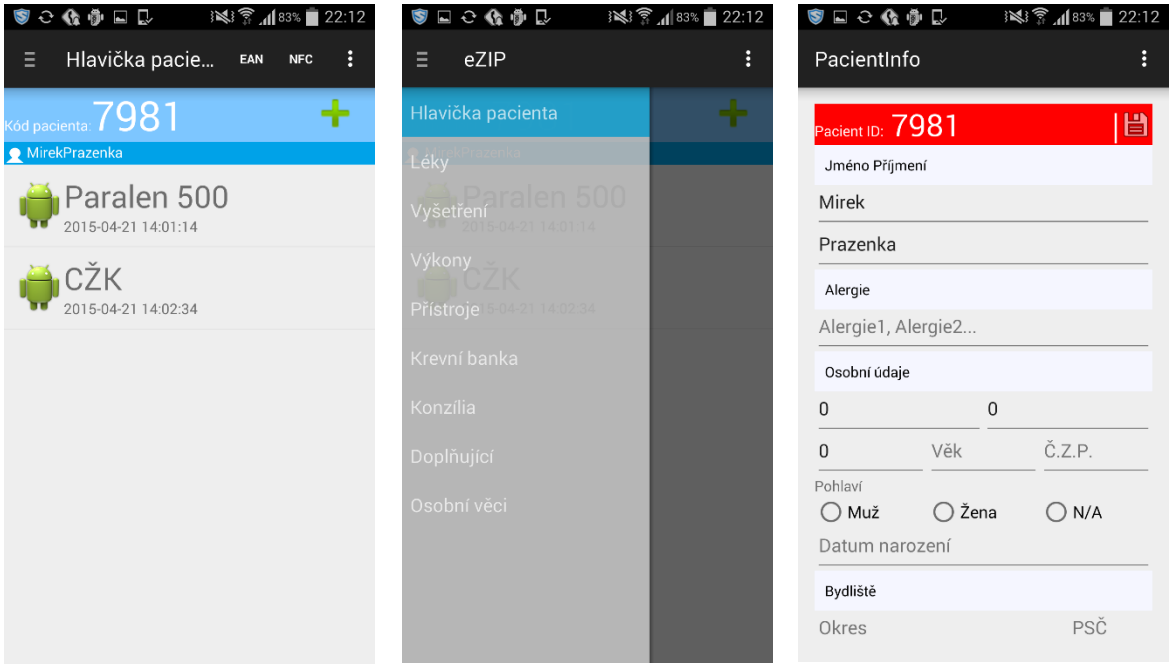
V této aktivitě se registruje nový pacient. Pacient získá své unikátní číslo. Dále se můžou doplnit osobní údaje pacienta, ale také nemusí. Tomu se tak děje z důvodu rychlé registrace. Stlačením NFC se objeví dialogové okno, které vyzve uživatele znovu potvrdit aktivitu. Potvrzením se pacientovo unikátní číslo zapíše do jeho tágu přiložením tágu k mobilnímu zařízení a přepne se do aktivity aktuálního stavu.

Aktuální stav

Zde se zobrazují aktuálně ošetřující pacienti. Zobrazují se jejich unikátní kód, jméno, příjmení, čas načtení a stopky. Stopky indikují čas ošetření a spouští se s vytvořením nové karty pacienta. Výběrem konkrétního pacienta se kliknutím zobrazí jeho záznam.

Databáze pacientů

Po ukončení záznamu se pacientská dokumentace ukládá do databáze. Zde jsou znázorněné časy načtení a ukončení pacienta a unikátní kód pacienta. Kliknutím na jednotlivého pacienta se objeví jeho dokumentace.



Obrázek 10:Uživatelské prostředí pro detail pacienta. Navigační řadič pro výběr záznamů a pacientské info pro dodatečnou editaci.

Po zvolení pacienta v aktuálním stavu se uživatel dostává na jeho záznam intenzivní péče. Záznam je tvořen formou časové linie. Horní řádek je vždy informační. V řádku se zobrazuje unikátní kód pacienta a jeho jméno. Zelený znak plus zobrazuje podrobnosti pacienta a umožni editovat osobní informace. Pod informačním řádkem se nachází prostor pro list view. Formou řádku se zde zobrazují jednotlivé záznamy. Tahem prstu z leva do pravá po displeji se objeví navigační obsah aplikace. Zde si uživatel smí navolit, jaký záznam ho zajímá a co chce evidovat. Horní lišta obsahuje popis aktuální aktivity a tři tlačítka. EAN a NFC slouží k načtení záznamu. EAN načítá data opticky pomocí kamery např. QR, EAN, Matrix kódy. NFC načítá kódy pomocí RFID a nevyužívá kamerové zařízení nýbrž modul NFC. Poslední tlačítko slouží k ukončení záznamu pacienta a přesunutí do databáze pacientů.

6.4.2 Funkcionalita

Hlavní funkcionalitu v této aplikaci tvoří záznamy pacientů. Je zapotřebí aplikaci naprogramovat tak, aby bylo možné vyplnit záznamy, které souvisí s konkrétním pacientem a konkrétní entitou. Další průvodní funkcionality aplikace

- Přihlášení se do aplikace (práce s interní databázi)
- Načítání jedno, dvoudimenzionálních kódu
- Generování unikátních kódu a zároveň jejich kontrola unikátností
- Práce s extérní databázi (Asynk Task)
- Chronometr pro měření čau pacienta stráveného na urgentním příjmu

6.5 Analýza identifikátoru

Analýzou identifikátoru se rozumí roztřídit jednotlivé entity na ty, které se budou moct načítat speciálním identifikačním kódem a na ty, které identifikátorem nejdou označit. Ty, které označit nejdou, jsou většinou veličiny např. tep, krevní tlak, teplota, saturace a další přístroji změřená data. Tyto data sice označit nejdou, ale dali by se ukládat na datový disk a využít pomocí internetu, jako automatický záznam k pacientu. Tím by se eliminovali chyby lidského faktoru, ale naproti tomu by to muselo mít vysoké nároky na bezpečnost plus systémové opatření. Ostatní entity, které nejdou označit a také nejsou diskrétně měřeny, nezbývá, než zapsat do systému ručně. Ty entity, které nejsou značeny a značit se můžou (např. vyšetření) se označí speciálním identifikačním číslem. Tedy každému druh vyšetření se přidělí jeho identifikační kód.[2]

6.5.1 Typy identifikátoru

Ne každý prvek je označený stejným typem identifikátoru. Léky jsou nejčastěji značeny na krabičce svým čárovým EAN kódem. Vyskytují se, ale léky které jsou značeny QR kódem. Nejlepší možný způsob identifikátoru je však RFID. RFID je identifikátor komunikující pomocí frekvencí a svůj kód doslova vyzařuje, tudíž se nejrychleji čte a je vlastně minimálně náročný na obsluhu čtení. Identifikátory se čtou pomocí média (např. mobilní telefon, čtečka kódu…).[2]

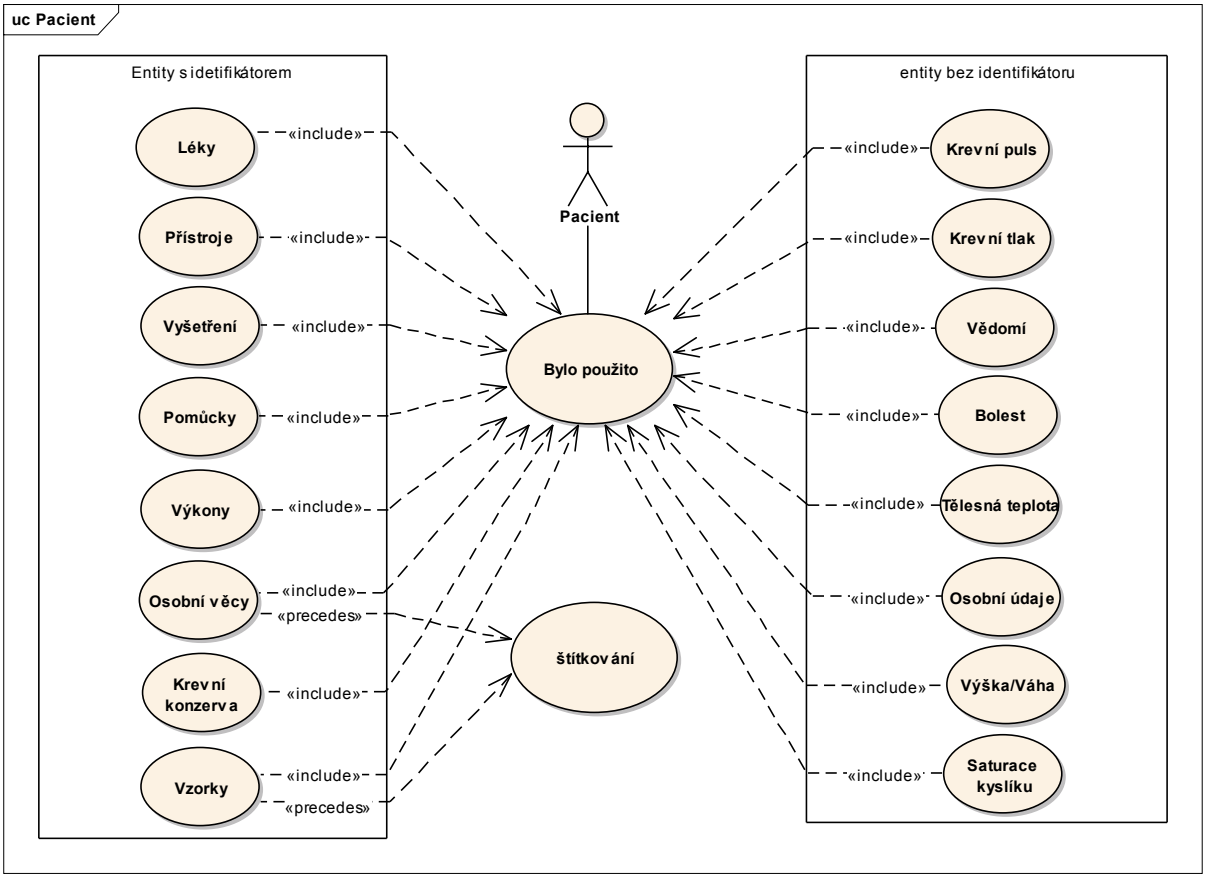


Diagram č. 5: *Roztřízeny informace o označených entitách.[2]*

7 Technické řešení

V této kapitole je popsáno technické řešení aplikace. Aplikace, jak už bylo zmíněno, je tvořena v Android Studiu jazykem JAVA. Aplikace byla tvořena dílčími způsoby. Android Studio je dostupné zdarma na stránkách https://developer.android.com/sdk/index.html. U instalace můžete být vyzváni nastavení proměnné PATH. Tj. nastavení cesty k instalovanému JDK. Pro Windows "Ovládací panely" -> "Systém" -> "Upřesnit" -> "Proměnné prostředí". Zde se upraví cesta ke staženému adresáři.

Android studio je poté připraveno k použití. Založení nového projektu Vás vyzívá k názvu projektu, dále k výběru platformy (mobil, tablet, wear, TV, glass). Při založení nového projektu v Android Studiu si můžete zvolit již předpřipravené aktivity. To se ocení při aplikaci, která pracuje s mapami nebo s postranním navigačním seznamem. Dále se jen aktivita pojmenuje a projekt se vytvoří. Vytvoří se kořenové struktura aplikace, kde hledáme složky „java“ a „res->layout“.

7.1 Čtení identifikačních kódu

Důležitou funkcionalitou pro sběr dat je čtení identifikačních kódů. Prvotním krokem pro optické čtení jednorozměrných a více rozměrných identifikátoru, je uvést potřebné řádky kódu v souboru „AndroidManifest.xml“.

```
<uses-permission android:name="android.permission.CAMERA"/>
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>

<uses-feature android:name="android.hardware.camera"/>
<uses-feature android:name="android.hardware.camera.autofocus"/>
```

Pro čtení NFC tagu, je zapotřebí do permissionu (závislostí) přidat NFC modul. Podobným způsobem, jako tomu bylo u 1D a 2D identifikátorech se souboru „AndroidManifest.xml“ zapíši tyto řádky.

```
<uses-permission android:name="android.permission.NFC" />
<uses-feature
    android:name="android.hardware.nfc"
    android:required="true" >
</uses-feature>
```

7.1.1 Zixing knihovna

Knihovnu Zxing vytvořila společnost Google. Je psaná v jazyku Java pro čtení a zápis 1D a 2D čárových kódů a je schopna rozpoznat kód bez komunikace se serverem. Také je nabízena v jiných portech, jako např. C nebo C++.[1]

Pro účely této aplikace bylo použito již hotových tříd volně dostupných zde:

- IntentIntengrator - zde se vytváří a definuje způsob čtení 1D a 2D (https://github.com/zxing/zxing/blob/master/android-integration/src/main/java/com/google/zxing/integration/android/IntentIntegrator.java)
- ImtemtResult – zde se zpracovává výsledek (https://github.com/zxing/zxing/blob/master/android-integration/src/main/java/com/google/zxing/integration/android/IntentResult.java)

Mírnou úpravou se výsledek zpracovává.

7.1.2 AsyncTask

Třída, která je v projektu využívaná velmi často, jak pro komunikaci s extérní databázi, tak pro čtení identifikačních prvků. Zjednodušeně se dá tvrdit, že se třída AsyncTask používá v případech, kdy je třeba provést operaci „na pozadí“ a nebrzdit chod programu „na popředí“.

Třídu AsyncTask tvoříme takto: „AsyncTask <params, progres, vysledek>“. Na pozici params jsou parametry, které metodě předáváme k výpočtu. Pozice progres, je pozice pro mezi výpočty (v projektu nevyužito proto volíme void) a posledním parametrem se definuje typ výsledku. AsyncTask prochází čtyřmi kroky:

- onPreExecute() – provádí operace před voláním vlákna.
- doInBackground(Params...) – přepne operace na pozadí jiného vlákna
- onProgressUpdate(Progress...) – aktualizuje operace z jakéhokoliv místa
- onPostExecute(Result) – nabídne výsledek na hlavním vlákně

Voláním těchto metod přepínáme vlákna procesoru. Zkráceně Asynctask se dá použít pomocí dvou kroků: doInBackground() a onPostExecute(d). AsyncTasks by měl být v ideálním případě použit pro krátké operace (několik sekund na nejvíce.).

7.1.3 NFC

Čtení NFC je tvořeno pomocí jedné třídy, která nejprve kontroluje, zda je NFC modu v zařízení zapnut (povolen). Pokud není, generuje se toustova hláška. Toust je buď krátkodobá (přibližně 1 sekundová) nebo dlouhodobá (5 sekundová) zpráva, která se uživateli zobrazí na displeji. Po zapnutí NFC adaptéru se čeká na událost. Událost je přiblížení se mobilního zařízení na vzdálenost schopnou přečíst obsah NFC tagu. Událostí se zavolá metoda pro čtení NFC. Tam se použije AsyncTask <tag, Void, String>“. Výsledek zpracováváme a pracujeme s databází. Dekódování informace obsažené v tagu, provádíme pomocí těchto řádků:

```
private String readText(NdefRecord record) throws UnsupportedEncodingException {

    byte[] payload = record.getPayload();
    String textEncoding = ((payload[0] & 128) == 0) ? "UTF-8" : "UTF-16";
    int languageCodeLength = payload[0] & 0063;
    return new String(payload, languageCodeLength + 1,
                        payload.length - languageCodeLength - 1, textEncoding);

}
```

Pokud je v tagu zapsaná informace, tak se zapíše do textového řetězce a vrací výsledek, který dále zpracováváme.

V případě, že budeme muset do NFC tagu zapisovat, bude zapotřebí si vytvořit NdefMessage. *NFC Data Exchange Format (NDEF) definuje formát zapouzdření zpráv pro výměnu informací mezi zařízeními respektujícími doporučení NFC fóra, tj. mezi dvěma aktivními NFC zařízeními, nebo aktivním zařízením a zařízením pasivním (tagem). Jedná se o binární formát zpráv, který může být použit k zapouzdření libovolných dat aplikačních protokolů do jedné zprávy.[20]*

```
private NdefMessage createNdefMessage(String content){

    NdefRecord ndefRecord =createTextRecord(content);
    NdefMessage ndefMessage = new NdefMessage(new NdefRecord[]{ndefRecord});

    return ndefMessage;
}
```

Zde jsme vytvořili NDEF zprávu, která obsahuje záznam. Záznam se tvoří v metodě createTextRecord().

```
private NdefRecord createTextRecord(String connect){
    try
    {
        byte[] language;
        language = Locale.getDefault().getLanguage().getBytes("UTF-8");
        final byte[] text = connect.getBytes("UTF-8");
        final int languageSize = language.length;
        final int textLenght = text.length;
        final ByteArrayOutputStream paylod = new ByteArrayOutputStream(1+ languageSize +
                textLenght);
        paylod.write((byte)(languageSize & 0x1F));
        paylod.write(language, 0, languageSize);
        paylod.write(text,0,textLenght);
        return  new NdefRecord(NdefRecord.TNF_WELL_KNOWN,NdefRecord.RTD_TEXT, new byte[0],
                paylod.toByteArray());
    }catch (UnsupportedEncodingException e){
        Log.e("CreateTextRecord",e.getMessage());
    }
    return null;
}
```

7.2 Ukládání dat

Android aplikace může ukládat data lokálně na mobilní zařízení nebo pomocí wifi sítě na externí databázový server. Při použití wifi sítě je zapotřebí připsat v AndroidManifestu.xml následující závisost:

```
<uses-permission android:name="android.permission.INTERNET" />
```

Z uložením dat se setkáváme v aplikaci již na začátku při přihlášení. Na zařízení je vytvořená tabulka uživatelů, kteří se zařízením pracují a mají přístupové údaje. Programově to znamená ošetřit aplikaci přihlašovacím jménem a heslem.

Další práce s daty se v aplikaci provádí v příjmu pacienta, kde ukládáme do externí databáze nového pacienta. Evidujeme jeho osobní data. V aktivitě pacient info tyto data můžeme editovat. V aplikaci vlastně k datům přistupujeme stále.

7.2.1 SQLite

Přihlášení do systému aplikace je prováděno pomocí SQLite databáze. Musíme brát v potaz podporované typy SQLite:

- TEXT (String v JAVA)
- INTIGER (Long v JAVA)
- REAL (Double v JAVA)

Ostatní typy se musí převést do jednoho z těchto typů. Kontrola datových typů v SQLite chybí. Pro přihlášení do systému byla vytvořena tabulka o třech atributech (id, přihlašovací jméno, heslo). Implementace v Android Studiu pomocí třídy MyDatabaseHelper, která dědí z SQLiteOpenHelper. `public class MyDatabaseHelper extends SQLiteOpenHelper {}` Do této třídy vložím proměnné s konstruktorem:

```
private static final String DATABASE_NAME = " userdatabase ";
private static final String TABLE_NAME = " uzivatel ";
private static final int DATABASE_VERSIONS = 1;
public static final String USER_ID = " uzivatel_id ";
public static final String USER_NAME = " uzivatel_name ";
public static final String USER_PASS = " uzivatel_pass ";

private static final String DATABASE_CREATE_USER = " CREATE TABLE " + TABLE_NAME + "(" +
        USER_ID + " INTEGER PRIMARY KEY AUTOINCREMENT, " +
        USER_NAME + " TEXT, " +
        USER_PASS + " TEXT);";

public MyDatabaseHelper(Context context) {
    super(context, DATABASE_NAME, null, DATABASE_VERSIONS);
    SQLiteDatabase db = this.getWritableDatabase();
}
```

V metodě onCreate() poté databázi vytvoříme a naplníme jedním uživatelem metodou insert:

```
@Override
public void onCreate(SQLiteDatabase db) {
    db.execSQL(DATABASE_CREATE_USER); //Vytvoření interní databázové tabuky uzivatel

    ContentValues cv = new ContentValues();
    cv.put(USER_ID, 1);
    cv.put(USER_NAME, "Nurse");
    cv.put(USER_PASS, 1111);

    db.insert(MyDatabaseHelper.TABLE_NAME,null,cv); //Vložení prvního záznamu do tabulky
}
```

Metoda insert() využívá třech parametrů:

- Název tabulky.
- Název jednoho sloupce tabulky. Pokud vkládáme celý záznam, píšeme null.
- Instance třídy ContentVaues, která obsahuje výchozí hodnoty sloupců řádků, které chceme do databáze vložit.[7]

Pro vyhledávání v tabulce se používá metoda query():

```
public Cursor getInformation(MyDatabaseHelper dop)
{
    SQLiteDatabase db = dop.getReadableDatabase();
    String[] columns = {USER_NAME,USER_PASS};
    return db.query(TABLE_NAME,columns, null, null, null, null, null);
}
```

Metoda query() obsahuje následující parametry z toho jen název tabulky je povinný. Zbytek parametrů lze doplnit hodnotou null:

- Název tabulky
- Seznam sloupců, které chceme z databáze vytáhnout.
- Podmínku dotazu definovanou pomocí WHERE, jež obsahuje parametry.
- Seznam hodnot, které mají parametry v klauzuli WHERE nahradit.
- Klauzuli GROUP BY.
- Klauzuli HAVING.
- Klauzuli ORDER BY.
- Klauzuli limit sloužící pro omezení počtu řádků vrácených dotazem.[7]

7.2.2 Externí databáze

Android od API úrovně 4.2 zakazuje přímé SQL dotazování. Doporučuje PHP nadstavbu nebo v lepším případě využití kódování a hashování. Tato část práce byla náročná z důvodu přístupu k externí databázi. Spojení s databází je realizováno přes PHP skripty. Soubory PHP jsou uloženy v adresáři public.html na adrese: https://homel.vsb.cz/~urb0063/diplomkaRFID/ + název PHP skriptu. V PHP skryptech byly prováděny tyto operace:

- Připojení k databázi
- Výběr dat z databáze
- Vložení dat do databáze
- Editování dat v databázi

Připojení k databázi přes **objektové PHP** se provádí například v textovém editoru, kde se vkládá tento PHP skript:

```php
<?php
$servername = "adresa.serveru.cz";
$username = "login";
$password = "heslo";
$dbname = "nazev_databaze";

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);
// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}
$conn->close();
?>
```

Výběr dat z databáze se provádí v PHP SQL dotazem, který je předán funkci query():

```php
$sql = "SELECT * FROM Pacienti WHERE cas_ukonceni is null ORDER BY cas_nacteni DESC";
$result = $conn->query($sql);
```

V proměnné result je výsledek výběru z databáze. Ten se převede do **formátu JSON**. JavaScript Object Notation (JavaScriptový objektový zápis, JSON) je způsob zápisu dat (datový formát) nezávislý na počítačové platformě, určený pro přenos dat, která mohou být organizována v polích nebo agregována v objektech. Vstupem je libovolná datová struktura (číslo, řetězec, boolean, objekt nebo z nich složené pole), výstupem je vždy řetězec.[20] Výsledek result převedeme na JSON zprávu, kterou vracíme a zobrazujeme na mobilním zařízení.

```php
$rows = array();
if ($result->num_rows > 0) {
    // output data of each row
    while($row = $result->fetch_assoc()) {
        $rows[] = $row;
    }
}else {
    echo "0 results";
}
```

Vložení dat do databáze se provádí přes PHP SQL příkazem insert (). SQL dotaz krok po kroku ("Vložení do tabulky (sloupec, sloupec, sloupec) hodnota (něco, něco, něco)"). Dále v PHP následuje podmínka pro kontrolu vloženého záznamu.

```php
$sql = "INSERT INTO Pacient_has_Leky (cas_nacteni, Leky_ZIP_id, Pacienti_id)
VALUES ('$time', '$a', '$pac')";

if (mysqli_query($conn, $sql)) {
    echo "New record created successfully";
} else {
    echo "Error: " . $sql . "<br>" . mysqli_error($conn);
}
```

Tento PHP skript se konkrétně využívá pro vytvoření záznamu o užití léku. Vytvoření záznamu léku je velice podobné ostatním evidujícím záznamům. Vždy se vychází z informace, která se snímá z tagu. V tagu je unikátní identifikační číslo entity. Podle kódu se vyhledá entita. Z entity se vybere id. Pomocí id se vytvoří záznam o užití.

```php
$KOD = $_POST["kodtag"];      // Informace z tagu
$time = date("Y-m-d H:i:s"); // Generování aktuálního času
$pac = $_POST["PacientID"];  // Pacientovo identifikační číslo
// SQL dotaz pro vyhledání id léku pode jeho kódu
$sql = "SELECT id FROM Leky_ZIP WHERE kod_lek=$KOD";
$result = $conn->query($sql);
if ($result->num_rows > 0) {
    // output data of each row
    while($row = $result->fetch_assoc()) {
        $a=$row["id"];      // Z řádku konkrétního léku vytažené id
}
```

Evidování dat z databáze využíváme v aktivitě pacient info, ale také množství léku v tabulce Pacient_has_Leky. Pro evidování množství léku byl vytvořen PHP skript.

```php
$zaznam_id = $_POST["idZaznamLeku"];
$mnozstvi = $_POST["mnozstvi"];

$sql = "UPDATE Pacient_has_Leky SET Mnozstvi_leku = '$mnozstvi' WHERE id='$zaznam_id'";
$result = $conn->query($sql);
```

Práce v Android Studiu

Operace nad databází se provádějí většinou v jiném vlákně, z důvodu nerušeného běhu aplikace. Zde se uplatňuje třída AsyncTask. Ta se spouští pomocí .execute() většinou v aktivitě, kde vyžadujeme data z extérní databáze.

```
new  GetAllCustomers().execute(new ApiConnector());


private  class  GetAllCustomers extends AsyncTask<ApiConnector,Long,JSONArray> {
    @Override
    protected JSONArray doInBackground(ApiConnector... params) {
        return params[0].GetAllCustomers();
    }
    @Override
    protected void onPostExecute(JSONArray jsonArray) {
        setListAdapter(jsonArray);
    }
}
```

V metodě doInBackground() předáme API Connector, což je třída, která byla vytvořena pro metody, které zprostředkovávají spojení s databází. Zde se již vrací JSONArray nebo JSONObject.

```
public JSONArray GetAllCustomers(){
    //Vložím zde URL, které bude pro aktuální pacienty
    String url = "https://homel.vsb.cz/~urb0063/diplomkaRFID/akt_pac.php";

    HttpEntity httpEntity = null;
    try{
        DefaultHttpClient httpClient = new DefaultHttpClient();
        HttpGet httpGet = new HttpGet(url);
        HttpResponse httpResponse = httpClient.execute(httpGet);
        httpEntity = httpResponse.getEntity(); // Odpověd z databáze
    } catch (ClientProtocolException e) {
        e.printStackTrace();
    } catch (IOException e) {
        e.printStackTrace();
    }
    //Převod HttpEntity na Jason zprávu
    JSONArray jsonArray = null;
    if(httpEntity != null){
        try{
            String entityResponse = EntityUtils.toString(httpEntity);
            jsonArray = new JSONArray(entityResponse);
        } catch (IOException e) {
            e.printStackTrace();
        } catch (JSONException e) {
            e.printStackTrace();
        }
    }
    return jsonArray;
}
```

Je důležité vědět, co se z databáze vrací. V našem případě existují dvě možnosti formátu dat. Víme, že se jedná o JSON ale podle obsahu dat se JSON děli na JSONObject a JSONArray. Rozdíl je v syntaxi. **JSONObject** obsahuje jeden řádek tabulky a zapisuje se do složených závorek:

{"id":"1","rc":"0","jmeno":"cjcjfjf","prijmeni":"xbdjjrjtfj,"cas_nacteni":"2015-04-01 13:29:37","kod_pacienta":"2960","cas_ukonceni":"2015-04-04"}

JSONArrays může obsahovat více objektů a zapisuje se do hranatých závorek:

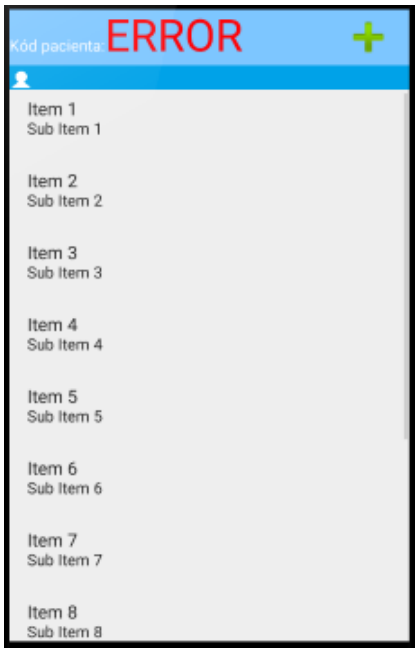
[{"id":"1","rc":"0","jmeno":"cjcjfjf","prijmeni":"xbdjjrjtfj,"cas_nacteni":"2015-04-01 13:29:37","kod_pacienta":"2960","cas_ukonceni":"2015-04-04}, {"id":"22293","rc":null, "jmeno":null,"prijmeni":null,"cas_nacteni":"2015-06-20 11:11:11", "kod_pacienta":"1234", "cas_ukonceni":"2015-04-08 18:36:27" }]

Tyto řetězce pak vidíme v třídě AsyncTasku metodě onPostExecute(), kde jsou zpracovány podle využití.

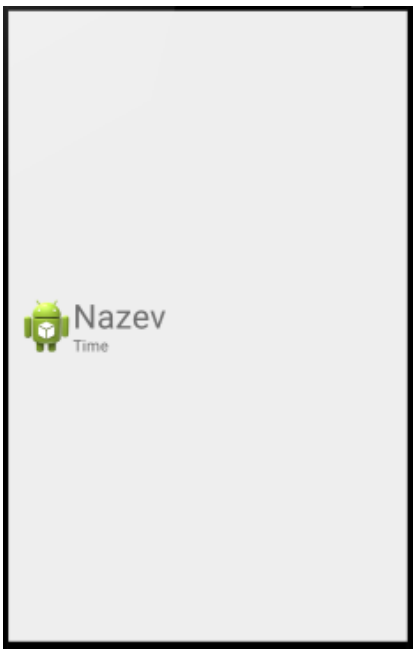
7.2.3 ListView

ListView je pohled, dědící ze skupiny AdapterView, který zobrazuje skupinu pohledu ve vertikálním rolovacím seznamu. Pro přenos různé skupiny dat využívá adaptéru. BaseAdapter vytváří pohled položky v seznamu. Obsahuje svůj grafický layout a stará se o vyplnění daty.

Prvním krokem si vytvoříme ListView, jako grafický prvek a přidělíme mu id. Pote vytvoříme layout i ro adapter. Ten bude definovat, jak bude položka v seznamu vypadat:



Obrázek 12: ListView seznam položek



Obrázek 11: Layout pro vzhled položky v ListView

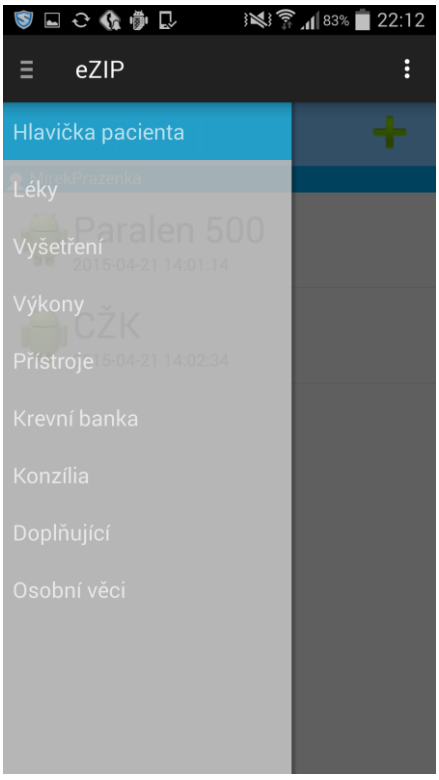
Po vytvoření layoutu se v požadované aktivitě najde ListView a zavolá se adaptér.

```
this.listViewZaznam = (ListView) findViewById(R.id.listViewZaznam);
zaznamAdapter = new GetZaznamyAdapter(this, 0, PacientID, this);
listViewZaznam.setAdapter(zaznamAdapter);
```

V adaptéru se vytváří třída AsyncTask pro dotazování k databázi. Výsledek se zpracuje a vyplní položky v seznamu. Příklad použití na stránkách: http://www.codelearn.org/android-tutorial/android-listview.

7.2.4 NavigationDrawer

NavigationDrawer je uzpůsobená aktivita, která využívá výsuvného fragmentu, který plní úkol postranního menu. Kliknutím některé z položek dokáže přepínat fragmenty nebo i aktivity. Android studio nabízí již předpřipravenou aktivitu. V práci byla tato aktivita využita pro přehled konkrétních tříd záznamu (léky, přístroje, vyšetření atd.). Aktivita se rozšíří o požadovaný počet položek v metodě onSectionAttached(int number) která vrací číslo pozice. V adresáři (res- value-string) se pak položky pojmenují.

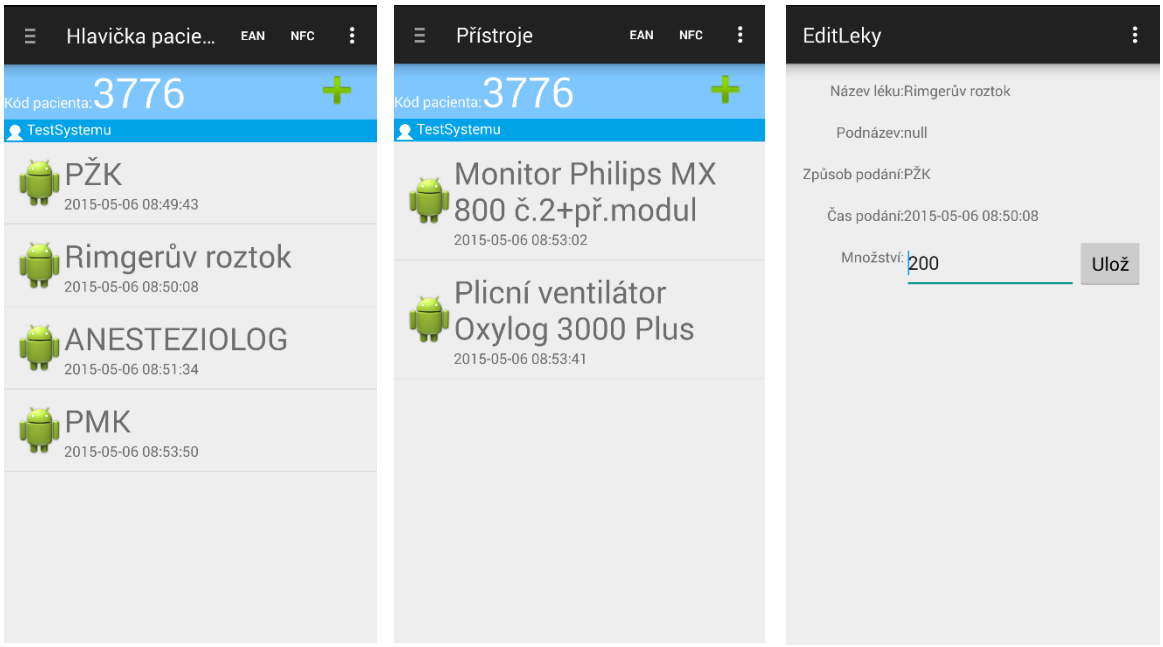


Obrázek 13: NavigationDrawer

8 Testování

Testování informačního systému proběhlo 6. 5. 2015 na urgentním příjmu FNO. Testování bylo prováděno pomocí jednoho mobilního zařízení s připojením na WIFI síť a s NFC hardwarem. Pracoviště urgentního příjmu bylo uzpůsobené pro sběr dat pomocí mobilního zařízení. K dispozici bylo zcela zajištěné (polepené NFC tagy) trauma lůžko II.

Testovací záznam byl uskutečněn na pacientovi s prvotní diagnózou krvácení do mozku. Pacient byl muž středního věku. Na urgentní příjem FNO byl převezen z jiné nemocnice v bezvědomí. Pacient byl v kritickém stavu, z toho důvodu byl indikován na urgentní příjem FNO. Zdravotnický personál dále kontaktoval neurologické oddělení FNO. Pacient u sebe neměl občanský ani zdravotnický průkaz. Jeho totožnost byla známá, avšak z důvodu ochrany osobních důvodů nezveřejněna (bez souhlasu pacienta). Registrace pacienta byla omezena, jako jméno bylo zadáno Test Systému. Další informace: Alergie – Nezjištěno, Výška – 180, Váha – 80, Zdravotní pojišťovna – 111. Registrace byla provedena 08:47:00 a trvala cca 2 minuty. Zdravotnický tým převzal pacienta a umístil ho na pacientské lůžko lokalizované, jako trauma lůžko II. Pacientovi byl v čase 08:49:43 zaveden žilní katetr. V záznamu vedený pod zkratkou PŽK. Dále po zavedení katetru mu byla podána v čase 8:50:08 infuze s názvem Rimgerův roztok s obsahem 200ml. K pacientovi v čase 8:51:34 byl přivolán anesteziolog. Zdravotní tým pacienta připojil k monitoru životních funkcí Philips MX800 č.2 + př. Modul. Anesteziolog napojil pacienta na plicní ventilátor Oxylog 3000 Plus. V dokumentu je vedený záznam o užití přístroje v čase 08:53:41 a výkon PMK 08:53:50. Pacient dále čekal na uvolnění místa na neurochirurgii a v čase 09:02:29 opustil urgentní příjem.



Obrázek 14: eZIP testovacího pacienta.

8.1 Zhodnocení

Zhodnocení výsledku testování je zavádějící. Systém nebyl testován na plné škále jeho funkčnosti. Testování bylo doprovázeno standartní papírovou dokumentací. Systém byl testován na jednom pacientovi. Výsledné záznamy se porovnaly a byli vyhodnocené rozdíly. Z požadavku víme, že systém by měl dodržet evidovanou škálu informací papírové formy. Bohužel systém nedokáže evidovat záznamy životních funkcí. Do budoucna se počítá se sdílením dat z monitoru životních funkcí, čímž by se vytvořil kompletní záznam intenzivní péče. V práci se nepodařilo vytvořit komunikační protokol z důvodu časové, bezpečnostní a technické úrovně složitosti. Jiné záznamy systém zvládá evidovat bez časových prodlev a přehledně. Uživatelská manipulace se systémem se jeví jako vyhovující. Průběh testování nevykazoval závažnějších problému. Závěrem zhodnocení testování je nevyhovující pro momentální nasazení do ostrého provozu. Testováním bylo zjištěno několik nedostatku, které se do budoucna musí odstranit.

8.1.1 Porovnání dokumentací

Papírová dokumentace eviduje kompletní záznam. Hlavní rozdíl je absence životních funkcí v elektronické verzi. Dále v elektronické verzi chybí údaje o délce infuze a umělé plicní ventilaci. Chybí stav vědomí pacienta tzv. RAMSAY úroveň 6 (hluboké bezvědomí). Dále chybí podpisy lékaře a evidující sestry.

Elektronická verze disponuje strukturovaným zápisem evidovaných dat. Evidence záznamu byla provedena v přesných časových okamžicích. Rychlý přehled a zápis dat s širší informační hodnotou. Například podrobnosti léku. Možnost dalších funkcionalit.

8.1.2 Chyby

Při testování vznikla celá řada chyb, která omezila výsledný elektronický záznam. Pro další zpracování se chyby evidovali, aby se ze systému eliminovali. V průběhu testování byli pozorovány tyto chyby:

- Aplikace padá, když v aktivitě aktuálního stavu není žádný pacient.
- Aplikace padá, když se uživatel snaží zapsat množství léku všude jinde, než v hlavičce pacienta.
- V aplikaci není prostor pro vytvoření poznámek.
- Při načítání záznamu EAN, NFC nefunguje kontrola správného načtení.
- Po načtení je potřeba aktualizovat obsah.
- Některé třídy nejsou uzpůsobený pro evidenci záznamu (Osobní věci, Doplňující, Krevní banka)
- Nefunkční přihlášení lékaře (bez možnosti navýšení uživatelů)

- Aplikace není ošetřena pro výpadek WIFI sítě
- Validaci unikátního čísla pacienta funguje, avšak aplikace při shodě kódu padá.

Tyto chyby se zpracovávají a pracuje se na jejich odstranění ze systému.

9 Závěr

Cílem této diplomové práce bylo navrhnout a implementovat informační systém, který je schopný evidovat elektronické záznamy intenzivní péče na urgentním příjmu fakultní nemocnice Ostrava. Informační systém je vytvořen pro mobilní zařízení s platformu Android. Informační systém sleduje koncept *Internet Of Things*. V diplomové práci byla řešena mobilní aplikace, která umožňuje čtení identifikačních kódu (EAN, QR, NFC) a ukládá data do databáze. Dále nabízí přehledné uživatelské rozhraní. Tato diplomová práce navazuje na bakalářskou práci s názvem „*Databáze pro evidenci označených entit sledující koncept Internet of Things"*, která řeší databázi k evidenci záznamu.

Diplomová práce vyžaduje dobré znalosti programovacího jazyku JAVA a zkušeností s programováním pro Android. Vyžaduje také základní znalosti databázových systému s PHP nadstavbou a dotazovacího jazyka SQL. Obsahem diplomové práce je návrh mobilní aplikace od analýzy prostředí po testování aplikace na urgentním příjmu. Výstup diplomové práce se nadále bude vyvíjet pro reálné využití v praxi.

Testování prokázalo dobré a rychlé uživatelské prostředí, které není náročné na obsluhu. Systém je schopen evidovat entity označené identifikačním kódem. Mobilní aplikace uchovává data v externí databázi, tudíž je náchylná na WIFI připojení. V rámci projektu lokalizace pozice pacienta se wifi připojení optimalizovalo a nyní nepředstavuje přímé riziko pro chod aplikace. Pro vyšší bezpečnostní opatření se doporučuje záznamy přechodně ukládat na vnitřní databázi mobilního zařízení za použití SQLite. Momentálně se na takovém řešení pracuje. Mobilní aplikace umožňuje sejmutím identifikačního kódu, vytvořit záznam o užití dané entity v databázi a následně tento záznam i zobrazit. V této práci technické řešení vyžaduje znát objekt entity, tak aby byl načtený záznam uložen do správné tabulky. Při testování se to prokázalo, jako nadbytečný krok. Navrhuje se řešení se společnou tabulkou všech záznamů s unikátním kódem. Mnoho nedostatku v tuto chvíli neumožňuje systém plně implementovat. Nedostatky vyplývající z testování a budou v nejbližší době eliminovány. Testování vypovídá o budoucích možnostech elektronické evidence záznamu intenzivní péče.

Mobilní aplikace běží na komerčně dostupném zařízení značky Samsung S4 s procesorem 1.9MHz, RAM pamětí 2048MB a vnitřní pamětí 16GB. Mobilní aplikace je tvořena Android Studiem verzí 1.1.0. Zpráva dat je prováděná MySQL Workbench 6.3 CE. Pro zprávu souboru na serveru byl použit software WinSCP. PHP skripty byli psány v textovém editoru.

Seznam použité literatury

[1] GREGOROVÁ, Hana. *Mobilní technologie s podporou konceptu internet of things*. Ostrava, 2013. Dostupné z: http://hdl.handle.net/10084/98878. Bakalářská práce. Vysoká škola báňská - Technická univerzita Ostrava. Fakulta elektrotechniky a informatiky. Vedoucí práce Jirka, Jakub.

[2] URBANCZYK, Tomáš. *Databáze pro evidenci označených entit sledující koncept Internet of Things*. Ostrava, 2013. Dostupné z: http://hdl.handle.net/10084/98907 . Bakalářská práce. Vysoká škola báňská - Technická univerzita Ostrava. Fakulta elektrotechniky a informatiky. Vedoucí práce Jirka, Jakub.

[3] ŠARMANOVÁ, Jana. VŠB – TECHNICKÁ UNIVERZITA OSTRAVA. *Databázové a informační systémy: E-learningové prvky pro podporu výuky odborných a technických předmětů*. první. Ostrava, 2007. CZ.O4.01.3/3.2.15.2/0326. ISBN 978-80-248-1499-5. Dostupné z: http://www.elearn.vsb.cz/archivcd/FEI/DAIS/DAIS.pdf

[4] JIRÁSKOVÁ, K. Správa pacientských dat v nemocničním informačním systému Clinicom. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2010. 39 s. Vedoucí bakalářské práce Ing. Helena Škutková.

[5] MISHA. *Databáze: Architektura databází* [online]. první. 2010 [cit. 2013-05-06]. Dostupné z: http://www.databaze.chytrak.cz/architektura.htm

[6] SOMMEROVÁ. *Základy RFID technologií: Výukový materiál* [online]. Ostrava [cit. 2013-01-11]. Dostupné na World Wide Web: <http://rfid.vsb.cz/miranda2/export/sites-root/rfid/cs/okruhy/informace/RFID_pro_Logistickou_akademii.pdf>. VŠB-TUO Hornicko-geologická fakulta.

[7] UJBÁNYAI, Miroslav. *Programujeme pro Android*. Vyd. 1. Praha: Grada, 2012, 187 s. Průvodce (Grada). ISBN 978-80-247-3995-3.

[8] WELLING, LUKE a Laura THOMSON. *MySQL: Průvodce základy databázového systému*. první. Brno: CP Books, a.s., 2005. ISBN 80-251-0671-3.

[9] VALTR. Příběh čárového kódu: Vznikl jen náhodou!. In: *Studentpoint* [online]. 2014 [cit. 2015-01-03]. Dostupné z: http://www.studentpoint.cz/244-studovna/16123-pribeh-caroveho-kodu-vznikl-jen-nahodou/#.VKgloiuG__8

[10] ŘEPA, V. *Analýza a návrh informačních systémů.* 1. vyd. Praha: Ekopress, 1999. 403 s. ISBN 80-86119-13-0

[11] DOLEŽAL, Luboš. *Dotkněte se RFID !: Úvod do technologie*. Praha, 2006 [online] [cit. 2013-01-11]. Dostupné na World Wide Web: http://www.rfid-epc.cz/dokumenty/

[12] WIKIPEDIA, otevřená encyklopedie. *Čárové kódy* [online]. Datum poslední revize 19.12.2012 [cit. 2014-28-12]. Dostupné na World Wide Web: http://cs.wikipedia.org/wiki/%C4%8C%C3%A1rov%C3%BD_k%C3%B3d

[13] DUBENG. *Čárové kódy* [online]. [cit. 2013-01-10]. Dostupné na World Wide Web: <http://www.duben.org/skola/fel/5.rocnik/NM/TypyKodu2D.htm>

[14] KRČMÁŘ, Petr. *QR kódy* [online]. Datum publikování 1. 3. 2010 [online] [cit. 2013-01-10]. Dostupné na World Wide Web: <http://www.root.cz/clanky/qr-kody-kilobajty-v-malem-obrazku/>

[15] WIKIPEDIE. *QR kód* [online]. Datum poslední editace 9. 1. 2013 [online] [cit. 2013-01-10]. Dostupné na World Wide Web: http://cs.wikipedia.org/wiki/QR_k%C3%B3d

[16] SEMECKÝ, Vojtěch. Android Studio: Nové vývojové prostředí. *Zdroják.cz* [online]. 4.11.2013. [cit. 2015-04-22]. Dostupné z: http://www.zdrojak.cz/clanky/android-studio-nove-vyvojove-prostredi/

[17] MILOŠ ŠEDA.: Databázové systémy [online]. Dostupné na internetu < http://www.uai.fme.vutbr.cz/~mseda/DBS02_BS.pdf>. [cit. 2013-04-05]

[18] ŠIROKÝ, Jaromír. *Modelování a návrh informačních systémů: Logická reprezentace dat informačních systémů* [online]. 2006. vyd. Ostrava, 2006 [cit. 2015-04-30]. Dostupné z: http://homen.vsb.cz/~s1i95/ISVDAS/IS/IS_model_dat_1.htm

[19] VOGE, Lars. VOGELLA. *Android SQLite database and content provider: Using the Android SQLite Database*. Kindle Edition, 2014. Dostupné z: http://www.vogella.com/tutorials/AndroidSQLite/article.html

[20] Wikipedie: Otevřená encyklopedie: NFC Data Exchange Format [online]. c2013 [citováno 4. 05. 2015]. Dostupný z WWW: http://cs.wikipedia.org/w/index.php?title=NFC_Data_Exchange_Format&oldid=10852591

[21] Wikipedie*: Otevřená encyklopedie: JavaScript Object Notation* [online]. c2015 [citováno 5. 05. 2015]. Dostupný z WWW: <http://cs.wikipedia.org/w/index.php?title=JavaScript_Object_Notation&oldid=12278357>

Seznam obrázků

Obrázek 1: *Pasivní RFID tag* 2
Obrázek 2: *Princip činnosti pasivního RFID* 3
Obrázek 3: *Ukázka QR kódu* 5
Obrázek 4: *Jednorozměrný EAN-13 kód.* 6
Obrázek 5: *Přehled uživatelského prostředí po vytvoření nového projektu.* 8
Obrázek 6: *Relační model dat.* 10
Obrázek 7: Princip evidence přístroje do eZIP. 17
Obrázek 8: *Uživatelské prostředí pro přihlášení se do systému.* 23
Obrázek 9:Uživatelské prostředí pro registraci, aktuální a databázi pacientů. 24
Obrázek 10:Uživatelské prostředí pro detail pacienta 25
Obrázek 11: Layout pro vzhled položky v ListView 35
Obrázek 12: ListView seznam položek 35
Obrázek 13: NavigationDrawer 36
Obrázek 14: eZIP testovacího pacienta. 37

Seznam tabulek

Tabulka č. 1: *Datový slovník pro typ entity Pacienti* 20
Tabulka č. 2: *Změněna vazební tabulka Pacient_has_leky_ZIP* 20

Seznam diagramů

Diagram č. 1: *Diagram aktivity příjmu pacienta.* 14
Diagram č. 2: *ER diagram. Vztahy mezi tabulky.* 21
Diagram č. 3: *Diagram datových toků popisující vrchní hierarchii.* 22
Diagram č. 4: *DFD popisující práci a užití systému v praxi.* 22
Diagram č. 5*: Roztřízeny informace o označených entitách.* 26

Seznam příloh

Příloha A:*Povolení výzkumu*…..…….…….. I
Příloha B: *Záznam intenzivní péče (první strana)*…… II
Příloha C: *Záznam intenzivní péče (druhá strana)*……..…….…... III
Příloha D: *Diagram práce z RFID*……..…...……...VI

Adresová struktura přiloženého DVD

- Diplomová práce
- Přílohy
- Diagramy
- Android Studio
- Mobilní aplikace

Příloha A – Povolení výzkumu

VŠB-TU Ostrava
Fakulta elektrotechniky a informatiky (FEI)
17. listopadu 15/2172, Ostrava-Poruba, 708 33
Tel.: +420 596 919 353
Sekretariat.fei@vsb.cz

Fakultní nemocnice Ostrava
17. Listopadu 1790
708 52 Ostrava-Poruba
tel.: +420 597 371 111
fno@fno.cz

Žádost o povolení výzkumu

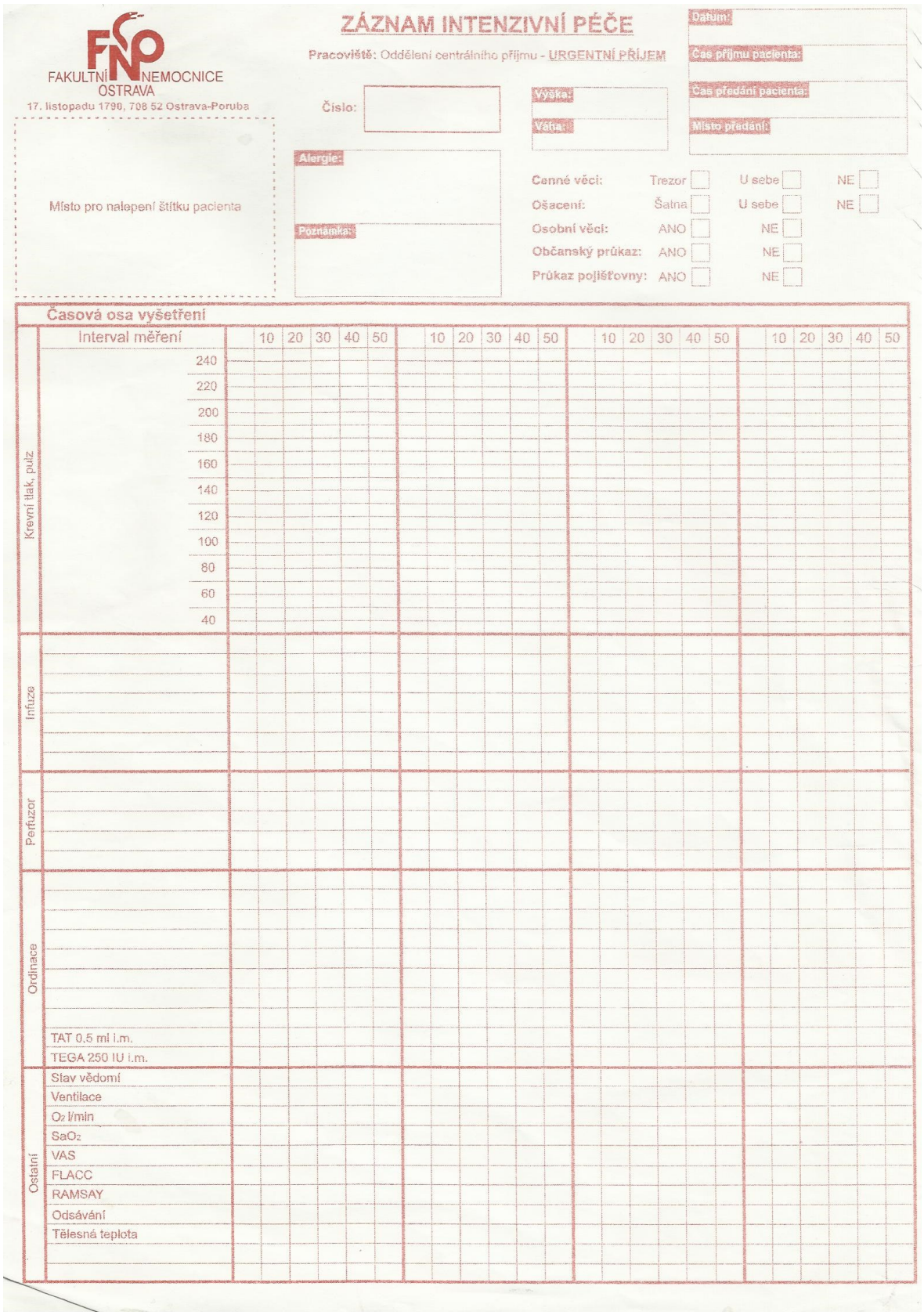
Já, **Bc. Tomáš Urbanczyk** student 5. ročníku Vysoké školy báňské technické univerzity Ostrava oboru biomedicínský inženýr, **žádám o povolení na urgentním příjmu**, abych získal podklady k vypracování diplomové práce s názvem: *Návrh a dílčí implementace elektronické verze Záznamu Intenzivní Péče.* Pod dohledem vedoucího práce doc. RNDr. Jindřichem Černohorským, CSc.

V Ostravě dne 30.9. 2014.

FAKULTNÍ NEMOCNICE OSTRAVA
Bc. Mária Dobešová
náměstek ředitele pro ošetřovatelskou péči
17. listopadu 1790, 708 52 Ostrava-Poruba

doc. RNDr. Jindřich Černohorský, CSc.	Bc. Tomáš Urbanczyk	Vedení FNO
Vedoucí práce	*Student*	

Příloha B-Záznam intenzivní péče (první strana)



FAKULTNÍ NEMOCNICE OSTRAVA
17. listopadu 1790, 708 52 Ostrava-Poruba

ZÁZNAM INTENZIVNÍ PÉČE

Pracoviště: Oddělení centrálního příjmu - URGENTNÍ PŘÍJEM

Místo pro nalepení štítku pacienta

Číslo:

Výška:

Váha:

Datum:

Čas příjmu pacienta:

Čas předání pacienta:

Místo předání:

Alergie:

Poznámka:

Cenné věci: Trezor ☐ U sebe ☐ NE ☐

Ošacení: Šatna ☐ U sebe ☐ NE ☐

Osobní věci: ANO ☐ NE ☐

Občanský průkaz: ANO ☐ NE ☐

Průkaz pojišťovny: ANO ☐ NE ☐

Časová osa vyšetření

Interval měření

10 20 30 40 50 | 10 20 30 40 50 | 10 20 30 40 50 | 10 20 30 40 50

Krevní tlak, puls

240
220
200
180
160
140
120
100
80
60
40

Infuze

Perfuzor

Ordinace

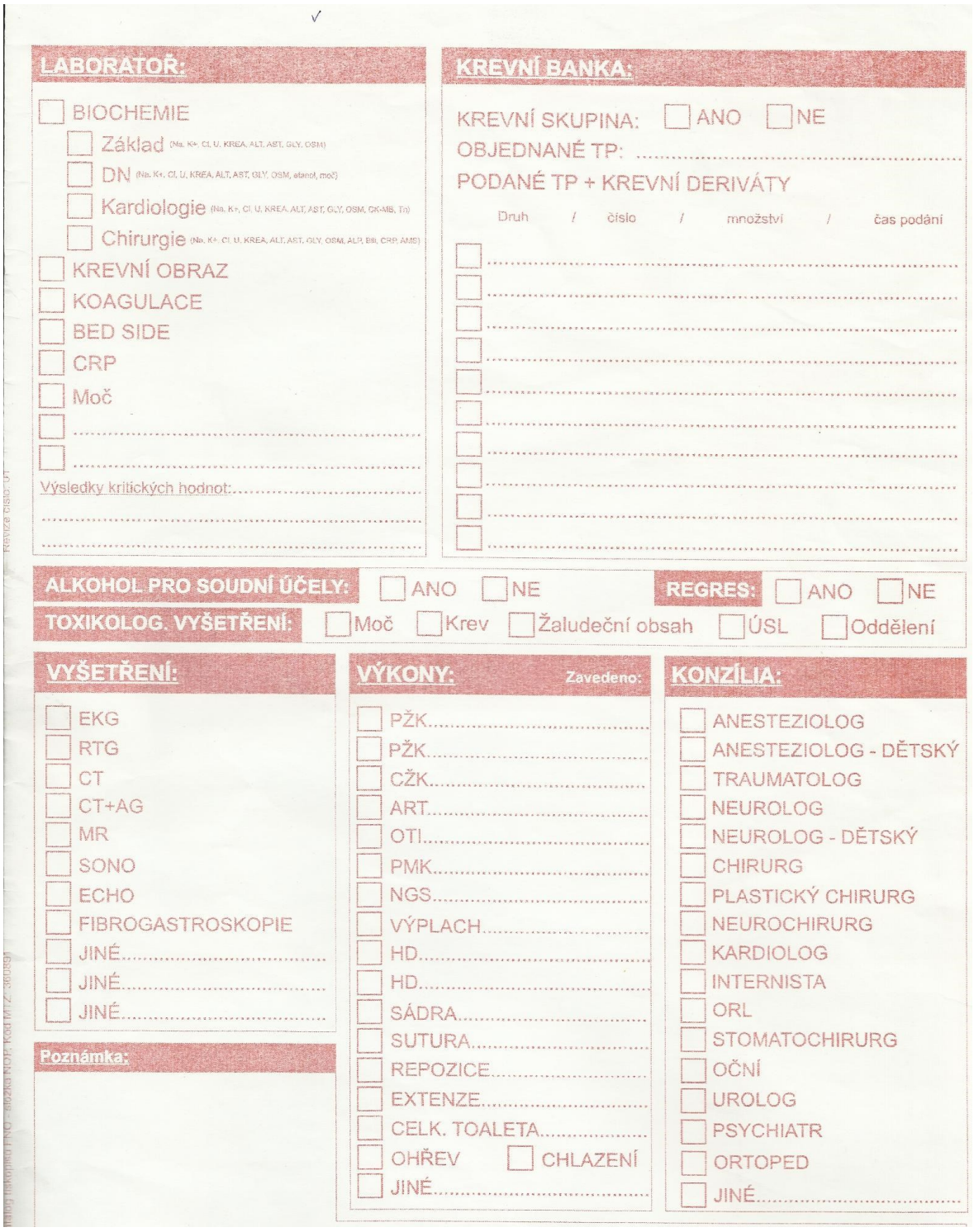
TAT 0,5 ml i.m.
TEGA 250 IU i.m.

Ostatní

Stav vědomí
Ventilace
O_2 l/min
SaO_2
VAS
FLACC
RAMSAY
Odsávání
Tělesná teplota

*Interní formulář fakultní nemocnice Ostrava (zdroj)

Příloha C-Záznam intenzivní péče (druhá strana strana)



*Interní formulář fakultní nemocnice Ostrava (zdroj)

Příloha D – Diagram práce s RFID